

Pembangunan Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Putra Firman Ardiansyah

NIM: 145150201111053



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PEMBANGUNAN APLIKASI PENJUALAN DAN PELAYANAN PELANGGAN PADA
PT. MEGA ABADI MOTORINDO

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Putra Firman Ardiansyah
NIM: 145150201111053

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Denny Sagita Rusdianto, S.Kom, M.Kom
NIP: 198511242015041001

Lutfi Fahani, S.Kom., M.T., M.Sc.
NIK: 2016078902171001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Desember 2018



Putra Firman Ardiansyah

NIM: 145150201111053

PRAKATA

Puji Syukur kepada Tuhan Yang Maha Esa karena berkat kasih dan karuniaNya penulis dapat menyelesaikan skripsi ini sebagai salah satu syarat untuk meraih gelar sarjana di Jurusan Informatika Fakultas Ilmu Komputer Universitas Brawijaya.

Judul yang penulis ajukan untuk skripsi ini adalah "Pembangunan Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo". Kelancaran dalam proses penelitian dan penulisan skripsi ini tidak lepas dari bimbingan, arahan, serta kerja sama dari berbagai pihak baik pada tahap persiapan, penyusunan, dan penyelesaian akhir skripsi ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada :

1. Orang tua dan adik penulis yang telah memberikan semangat, motivasi, doa, dan dukungan baik materil maupun non materil kepada penulis selama perkuliahan.
2. Bapak Denny Sagita R S.Kom, M.Kom selaku Pembimbing I, yang telah membimbing dan mengarahkan penulis dalam proses penyusunan laporan skripsi sehingga skripsi ini dapat terselesaikan dengan baik.
3. Bapak Lutfi Fanani, S.Kom., M.T., M.Sc. selaku Pembimbing II, yang telah membimbing dan mengarahkan penulis dalam proses penyusunan laporan skripsi sehingga skripsi ini dapat terselesaikan dengan baik.
4. Ibu Dian Widayanti selaku manager di PT. Mega Abadi Motorindo yang telah membantu penulis dalam pengumpulan data pada penelitian ini.
5. Oktavina Nurul Dwi Wijayati dan Afifah Salsabila yang telah memberikan semangat dan motivasi untuk segera menyelesaikan skripsi.
6. Nadia Kusuma P, Ericko Wicaksono dan teman-teman lain yang telah membantu penulis jika penulis mengalami masalah dalam pembuatan dokumen.
7. Seluruh pihak yang telah membantu kelancaran skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun akan penulis terima dengan senang hati untuk penyempurnaan dari segala kekurangan yang ada pada penelitian ini. Akhir kata, penulis harap skripsi ini dapat memberikan manfaat dan menambah wawasan bagi pembaca dan penulis sendiri.

Malang, 26 Desember 2018

Penulis

me@ardi.my.id

ABSTRAK

Putra Firman Ardiansyah, Pembangunan Aplikasi Penjualan Dan Pelayanan Pelanggan Pada PT. Mega Abadi Motorindo

Pembimbing: Denny Sagita Rusdianto, S.Kom, M.Kom dan Lutfi Fanani, S.Kom., M.T., M.Sc.

Dengan semakin berkembangnya perekonomian yang ditandai dengan adanya era globalisasi dan kemajuan teknologi yang pesat memungkinkan tersedianya sarana bagi pemenuhan kebutuhan pelanggan akan barang dan jasa menjadi lebih mudah. Situasi tersebut akan menimbulkan persaingan yang ketat antar perusahaan baik yang menghasilkan produk maupun jasa. Oleh karena itu, penjualan dan pelayanan pelanggan dalam suatu perusahaan menjadi kunci untuk bersaing. Untuk mempermudah melakukan pelayanan terhadap pelanggan seperti mengetahui info tentang promo, baik promo pembelian sepeda maupun pengingat waktu servis motor hal tersebut dapat dilakukan melalui layanan pesan sosial *WhatsApp* yang diintegrasikan dengan sistem. Hasil dari penelitian ini terdapat enam aktor, delapan puluh satu kebutuhan fungsional. Serta hasil pengujian yang seratus persen valid.

Kata kunci: rekayasa perangkat lunak, pelayanan, penjualan, website

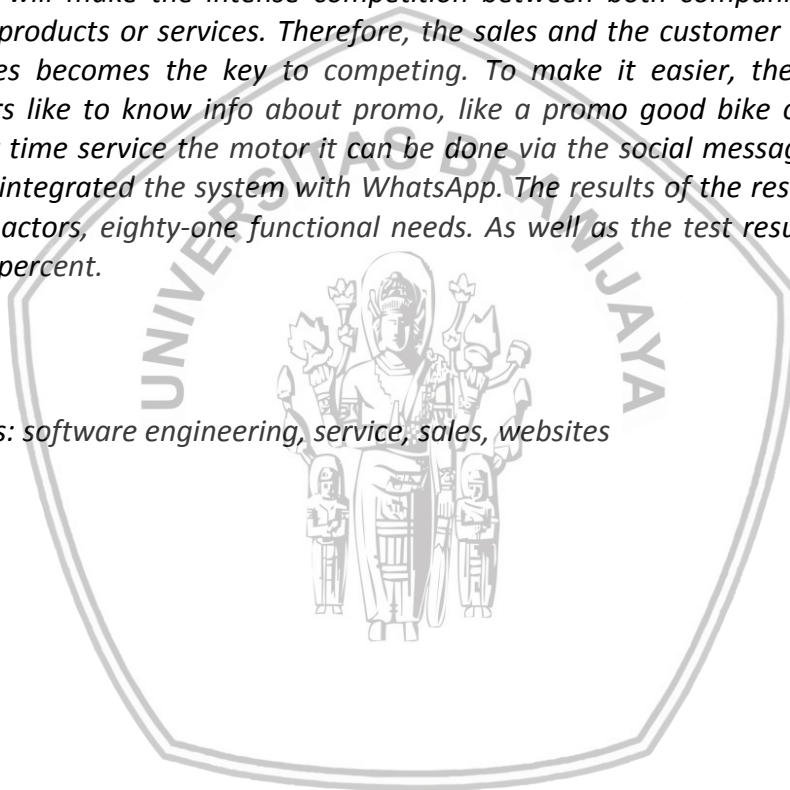
ABSTRACT

Putra Firman Ardiansyah, *Development of Customer Sales and Service Applications at PT. Mega Abadi Motorindo*

Supervisors: Denny Sagita Rusdianto, S.Kom, M.Kom and Lutfi Fanani, S.Kom., M.T., M.Sc.

With the development of the economy marked by the era of globalization and the rapid advancement of technology, allows the availability of means for the fulfillment of customer needs will goods and services becomes much easier. That situation will make the intense competition between both companies of them produce products or services. Therefore, the sales and the customer service in a companies becomes the key to competing. To make it easier, the service to customers like to know info about promo, like a promo good bike or purchase reminder time service the motor it can be done via the social messaging service that I've integrated the system with WhatsApp. The results of the research there were six actors, eighty-one functional needs. As well as the test results are one hundred percent.

Keywords: *software engineering, service, sales, websites*



DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xix
DAFTAR LAMPIRAN	xx
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Tinjauan Pustaka	4
2.2 Bahasa Pemrograman PHP	4
2.2.1 Definisi PHP	4
2.3 Laravel <i>Framework</i>	4
2.3.1 Definisi <i>Framework</i>	4
2.3.2 Definisi Laravel Framework.....	5
2.4 MySQL	6
2.4.1 Definisi MySQL	6
2.4.2 Fitur dan Kelebihan MySQL.....	6
2.5 Software Development Life Cycle	6
2.5.1 SDLC Waterfall.....	6
2.6 UML.....	8
2.6.1 Use Case Diagram	8

2.6.2 Sequence Diagram	9
2.6.3 Class Diagram	10
2.7 Pengujian	11
2.7.1 Whitebox testing.....	11
2.7.2 Blackbox testing	13
BAB 3 METODOLOGI PENELITIAN	14
3.1 Studi Literatur	14
3.2 Rekayasa Kebutuhan.....	15
3.3 Perancangan Sistem.....	15
3.4 Implementasi Sistem	16
3.5 Pengujian Sistem.....	16
3.6 Kesimpulan dan Saran	17
BAB 4 ANALISIS KEBUTUHAN SISTEM	18
4.1 Gambaran Umum Sistem.....	18
4.2 Analisis Kebutuhan	18
4.2.1 Elisitasi Kebutuhan.....	18
4.3 Identifikasi Aktor.....	21
4.4 Spesifikasi Kebutuhan.....	22
4.4.1 Daftar Kebutuhan Fungsional	22
4.5 Pemodelan Kebutuhan	31
4.5.1 Use Case Diagram	31
4.5.2 <i>Use Case Scenario</i>	31
4.6 Pembahasan Hasil Analisis Kebutuhan	56
BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM	57
5.1 Perancangan sistem	57
5.1.1 Pemodelan Sequence Diagram	57
5.1.2 Pemodelan <i>Class Diagram</i>	59
5.1.3 Perancangan Komponen	61
5.1.4 Perancangan Data	62
5.1.5 Perancangan Antarmuka Aplikasi	65
5.2 Implementasi Sistem	72
5.2.1 Spesifikasi Sistem	72

5.2.2 Batasan Implementasi.....	74
5.2.3 Implementasi Database	74
5.2.4 Implementasi Antarmuka	75
5.2.5 Pembahasan Hasil Implementasi	77
BAB 6 PENGUJIAN	78
6.1 Pengujian Unit.....	78
6.1.1 Pengujian Unit Klas InvoiceController Untuk Operasi LockInvoice	78
6.1.2 Pengujian Unit Klas PrintController Untuk Operasi PrintDeliveryOrder	79
6.1.3 Pengujian Unit Klas AutoComplateController Untuk Operasi SearchNoInvoiceWithTagihan	81
6.2 Pengujian Integrasi	83
6.2.1 Pengujian Integrasi Klas EmailController Untuk Operasi sendPermintaanBarangKePusat.....	83
6.2.2 Pengujian Integrasi Klas AutoComplateController Untuk Operasi searchNoPo	85
6.2.3 Pengujian Integrasi Klas CrmDashboardController Untuk Operasi cronTaskUltah	86
6.3 Pengujian Validasi	88
6.3.1 Pengujian Validasi Login.....	88
6.3.2 Pengujian Validasi Logout	89
6.3.3 Pengujian Validasi Melihat master data motor	90
6.3.4 Pengujian Validasi Membuat master data motor	90
6.3.5 Pengujian Validasi Mengubah master data motor	90
6.3.6 Pengujian Validasi Menghapus master data motor.....	91
6.3.7 Pengujian Validasi Melihat daftar harga beli	91
6.3.8 Pengujian Validasi Menambah daftar harga beli	92
6.3.9 Pengujian Validasi Mengubah daftar harga beli	92
6.3.10 Pengujian Validasi Menghapus daftar harga beli	93
6.3.11 Pengujian Validasi Melihat daftar harga jual	93
6.3.12 Pengujian Validasi Menambah daftar harga jual	93
6.3.13 Pengujian Validasi Mengubah daftar harga jual	94

6.3.14 Pengujian Validasi Menghapus daftar harga jual	94
6.3.15 Pengujian Validasi Melihat daftar permintaan barang.....	94
6.3.16 Pengujian Validasi Membuat permintaan barang baru.....	95
6.3.17 Pengujian Validasi Mengubah permintaan barang baru	95
6.3.18 Pengujian Validasi Menghapus permintaan barang baru.....	96
6.3.19 -Pengujian Validasi Mengirim informasi permintaan barang baru	96
6.3.20 Pengujian Validasi Membuat bukti penerimaan barang	97
6.3.21 Pengujian Validasi Melihat bukti penerimaan barang.....	97
6.3.22 Pengujian Validasi Mengubah bukti penerimaan barang.....	97
6.3.23 Pengujian Validasi Menghapus bukti penerimaan barang	98
6.3.24 Pengujian Validasi Mengunci status bukti penerimaan barang	98
6.3.25 Pengujian Validasi Melihat jumlah stok barang.....	99
6.3.26 Pengujian Validasi Membuat order penjualan baru	99
6.3.27 Pengujian Validasi Melihat order penjualan.....	99
6.3.28 Pengujian Validasi Mengubah order penjualan	100
6.3.29 Pengujian Validasi Menghapus order penjualan	100
6.3.30 Pengujian Validasi Mencetak detail order penjualan	101
6.3.31 Pengujian Validasi Membuat surat jalan	101
6.3.32 Pengujian Validasi Mengubah surat jalan.....	101
6.3.33 Pengujian Validasi Melihat surat jalan.....	102
6.3.34 Pengujian Validasi Menghapus surat jalan	102
6.3.35 Pengujian Validasi Mencetak surat jalan	102
6.3.36 Pengujian Validasi Membuat surat faktur pajak.....	103
6.3.37 Pengujian Validasi Mengubah surat faktur pajak	103
6.3.38 Pengujian Validasi Melihat surat faktur pajak	104
6.3.39 Pengujian Validasi Menghapus surat faktur pajak.....	104
6.3.40 Pengujian Validasi Mencetak surat faktur pajak.....	104
6.3.41 Pengujian Validasi Membuat invoice	105
6.3.42 Pengujian Validasi Mengubah invoice	105
6.3.43 Pengujian Validasi Melihat invoice	106
6.3.44 Pengujian Validasi Menghapus invoice.....	106

6.3.45 Pengujian Validasi Menambah sales order pada invoice	106
6.3.46 Pengujian Validasi Mencetak detail invoice.....	107
6.3.47 Pengujian Validasi Membuat master data pelanggan	107
6.3.48 Pengujian Validasi Mengubah master data pelanggan.....	108
6.3.49 Pengujian Validasi Melihat master data pelanggan.....	108
6.3.50 Pengujian Validasi Menghapus master data pelanggan	109
6.3.51 Pengujian Validasi Membuat master data leasing.....	109
6.3.52 Pengujian Validasi Mengubah master data leasing	109
6.3.53 Pengujian Validasi Melihat master data leasing	110
6.3.54 Pengujian Validasi Menghapus master data leasing	110
6.3.55 Pengujian Validasi Membuat data kas master.....	111
6.3.56 Pengujian Validasi Mengubah data kas master	111
6.3.57 Pengujian Validasi Melihat data kas master	112
6.3.58 Pengujian Validasi Menghapus data kas master	112
6.3.59 Pengujian Validasi Membuat data kas terima	112
6.3.60 Pengujian Validasi Mengubah data kas terima.....	113
6.3.61 Pengujian Validasi Melihat data kas terima.....	113
6.3.62 Pengujian Validasi Menghapus data kas terima	114
6.3.63 Pengujian Validasi Membuat data kas keluar	114
6.3.64 Pengujian Validasi Mengubah data kas keluar	114
6.3.65 Pengujian Validasi Melihat data kas keluar	115
6.3.66 Pengujian Validasi Menghapus data kas keluar	115
6.3.67 Pengujian Validasi Melihat riwayat kas terima.....	116
6.3.68 Pengujian Validasi Melihat riwayat kas keluar	116
6.3.69 Pengujian Validasi Melihat riwayat sales order	116
6.3.70 Pengujian Validasi Menambah data antrian servis.....	117
6.3.71 Pengujian Validasi Mengubah data antrian servis.....	117
6.3.72 Pengujian Validasi Melihat data antrian servis.....	118
6.3.73 Pengujian Validasi Menghapus data antrian servis	118
6.3.74 Pengujian Validasi Merubah status antrian servis.....	118
6.3.75 Pengujian Validasi Melakukan pengecekan gratis servis.....	119
6.4 Pembahasan Hasil Pengujian	119

6.4.1 Analisis Pengujian Unit.....	119
6.4.2 Analisis Pengujian Integrasi.....	119
6.4.3 Analisis Pengujian Validasi	120
BAB 7 Penutup	121
7.1 Kesimpulan.....	121
7.2 Saran	122
DAFTAR REFERENSI	123
LAMPIRAN A HASIL WAWANCARA.....	124



DAFTAR TABEL

Tabel 2.1 Simbol simbol pada <i>Use Case Diagram</i>	8
Tabel 2.2 Simbol simbol pada <i>sequence diagram</i>	9
Tabel 2.3 Simbol simbol pada <i>class diagram</i>	10
Tabel 2.4 Tingkatan angka cyclomatic complexity	13
Tabel 4.1 Identifikasi Aktor	21
Tabel 4.2 Daftar Kebutuhan Fungsional dan Pemetaan Nama Use Case	22
Tabel 4.3 <i>Use Case Scenario</i> Login	31
Tabel 4.4 <i>Use Case Scenario</i> Logout	33
Tabel 4.5 <i>Use Case Scenario</i> melihat master data motor	33
Tabel 4.6 <i>Use Case Scenario</i> membuat master data motor	33
Tabel 4.7 <i>Use Case Scenario</i> mengubah master data motor	33
Tabel 4.8 <i>Use Case Scenario</i> menghapus master data motor	34
Tabel 4.9 <i>Use Case Scenario</i> melihat daftar harga beli	34
Tabel 4.10 <i>Use Case Scenario</i> menambah daftar harga beli	34
Tabel 4.11 <i>Use Case Scenario</i> mengubah daftar harga beli	35
Tabel 4.12 <i>Use Case Scenario</i> menghapus daftar harga beli	35
Tabel 4.13 <i>Use Case Scenario</i> melihat daftar harga jual	35
Tabel 4.14 <i>Use Case Scenario</i> menambah daftar harga jual	35
Tabel 4.15 <i>Use Case Scenario</i> mengubah daftar harga jual	36
Tabel 4.16 <i>Use Case Scenario</i> menghapus daftar harga jual	36
Tabel 4.17 <i>Use Case Scenario</i> melihat daftar permintaan barang	36
Tabel 4.18 <i>Use Case Scenario</i> membuat permintaan barang baru	37
Tabel 4.19 <i>Use Case Scenario</i> mengubah permintaan barang baru	37
Tabel 4.20 <i>Use Case Diagram</i> menghapus permintaan barang baru	37
Tabel 4.21 <i>Use Case Scenario</i> mengirim informasi permintaan barang	38
Tabel 4.22 <i>Use Case Scenario</i> membuat bukti penerimaan barang	38
Tabel 4.23 <i>Use Case Scenario</i> melihat bukti penerimaan barang	38
Tabel 4.24 <i>Use Case Scenario</i> mengubah bukti penerimaan barang	39
Tabel 4.25 <i>Use Case Scenario</i> menghapus bukti penerimaan barang	39
Tabel 4.26 <i>Use Case Scenario</i> mengunci status bukti penerimaan barang	39

Tabel 4.27 <i>Use Case Scenario</i> melihat jumlah stok barang	39
Tabel 4.28 <i>Use Case Scenario</i> membuat order penjualan baru.....	40
Tabel 4.29 <i>Use Case Scenario</i> melihat order penjualan	40
Tabel 4.30 <i>Use Case Scenario</i> mengubah order penjualan	40
Tabel 4.31 <i>Use Case Scenario</i> menghapus order penjualan.....	41
Tabel 4.32 <i>Use Case Scenario</i> mencetak order penjualan.....	41
Tabel 4.33 <i>Use Case Scenario</i> membuat surat jalan	41
Tabel 4.34 <i>Use Case Scenario</i> melihat surat jalan	42
Tabel 4.35 <i>Use Case Scenario</i> mengubah surat jalan	42
Tabel 4.36 <i>Use Case Scenario</i> menghapus surat jalan.....	42
Tabel 4.37 <i>Use Case Scenario</i> mencetak surat jalan.....	42
Tabel 4.38 <i>Use Case Scenario</i> membuat surat faktur pajak	43
Tabel 4.39 <i>Use Case Scenario</i> melihat surat faktur pajak.....	43
Tabel 4.40 <i>Use Case Scenario</i> mengubah surat faktur pajak.....	43
Tabel 4.41 <i>Use Case Scenario</i> menghapus surat faktur pajak.....	44
Tabel 4.42 <i>Use Case Scenario</i> mencetak surat faktur pajak	44
Tabel 4.43 <i>Use Case Scenario</i> membuat invoice	44
Tabel 4.44 <i>Use Case Scenario</i> melihat invoice	45
Tabel 4.45 <i>Use Case Scenario</i> mengubah invoice	45
Tabel 4.46 <i>Use Case Scenario</i> menghapus invoice	45
Tabel 4.47 <i>Use Case Scenario</i> mencetak invoice	45
Tabel 4.48 <i>Use Case Scenario</i> menambah sales order pada invoice	46
Tabel 4.49 <i>Use Case Scenario</i> membuat master data pelanggan.....	46
Tabel 4.50 <i>Use Case Scenario</i> melihat master data pelanggan	46
Tabel 4.51 <i>Use Case Scenario</i> mengubah master data pelanggan	47
Tabel 4.52 <i>Use Case Scenario</i> menghapus master data pelanggan.....	47
Tabel 4.53 <i>Use Case Scenario</i> membuat master data leasing	47
Tabel 4.54 <i>Use Case Scenario</i> melihat master data leasing.....	48
Tabel 4.55 <i>Use Case Scenario</i> mengubah master data leasing.....	48
Tabel 4.56 <i>Use Case Scenario</i> menghapus master data leasing	48
Tabel 4.57 <i>Use Case Scenario</i> membuat data kas master	48
Tabel 4.58 <i>Use Case Scenario</i> melihat data kas master.....	49

Tabel 4.59 <i>Use Case Scenario</i> mengubah data kas master	49
Tabel 4.60 <i>Use Case Scenario</i> menghapus data kas master	49
Tabel 4.61 <i>Use Case Scenario</i> membuat data kas terima	50
Tabel 4.62 <i>Use Case Scenario</i> melihat data kas terima	50
Tabel 4.63 <i>Use Case Scenario</i> mengubah data kas terima	50
Tabel 4.64 <i>Use Case Scenario</i> menghapus data kas terima	50
Tabel 4.65 <i>Use Case Scenario</i> membuat data kas keluar	51
Tabel 4.66 <i>Use Case Scenario</i> melihat data kas keluar	51
Tabel 4.67 <i>Use Case Scenario</i> mengubah data kas keluar	51
Tabel 4.68 <i>Use Case Scenario</i> menghapus data kas keluar	52
Tabel 4.69 <i>Use Case Scenario</i> melihat riwayat kas terima	52
Tabel 4.70 <i>Use Case Scenario</i> melihat riwayat kas keluar	52
Tabel 4.71 <i>Use Case Scenario</i> melihat riwayat sales order	52
Tabel 4.72 <i>Use Case Scenario</i> membuat data antrian servis	53
Tabel 4.73 <i>Use Case Scenario</i> melihat data antrian servis	53
Tabel 4.74 <i>Use Case Scenario</i> mengubah data antrian servis	53
Tabel 4.75 <i>Use Case Scenario</i> menghapus data antrian servis	54
Tabel 4.76 <i>Use Case Scenario</i> merubah status antrian servis	54
Tabel 4.77 <i>Use Case Scenario</i> melakukan pengecekan gratis servis	54
Tabel 4.78 <i>Use Case Scenario</i> melihat riwayat pesan kepada pelanggan	54
Tabel 4.79 <i>Use Case Scenario</i> Mengirim pesan kepada pelanggan	55
Tabel 4.80 <i>Use Case Scenario</i> Mengirim pesan kepada banyak pelanggan	55
Tabel 4.81 <i>Use Case Scenario</i> Mengirim pesan otomatis pengingat servis	55
Tabel 4.82 <i>Use Case Scenario</i> Mengirim pesan otomatis ucapan ulang tahun	56
Tabel 4.83 <i>Use Case Scenario</i> mengunci status invoice	56
Tabel 5.1 Struktur Tabel master_pelanggan	64
Tabel 5.2 Struktur Tabel sales_order	64
Tabel 5.3 Struktur Tabel faktur_penjualan	65
Tabel 5.4 Struktur Tabel invoice	65
Tabel 5.5 Struktur Tabel kas_master	65
Tabel 5.6 Penjelasan antarmuka sales order	66
Tabel 5.7 Penjelasan antarmuka list invoice	67

Tabel 5.8 Penjelasan antarmuka master pelanggan	68
Tabel 5.9 Penjelasan antarmuka master pelanggan	70
Tabel 5.10 Penjelasan antarmuka kas master	71
Tabel 5.11 Penjelasan antarmuka penerimaan barang	72
Tabel 5.12 Spesifikasi Perangkat Keras	73
Tabel 5.13 Spesifikasi Perangkat Lunak	73
Tabel 5.14 Spesifikasi Sistem Operasi	74
Tabel 6.1 Hasil Pengujian Unit Kelas InvoiceController Untuk Operasi LockInvoice	79
Tabel 6.2 Hasil Pengujian Unit Kelas PrintController Untuk Operasi PrintDeliveryOrder	81
Tabel 6.3 Hasil Pengujian Unit Kelas AutoComplateController Untuk Operasi SearchNoInvoiceWithTagihan	82
Tabel 6.4 Hasil Pengujian Integrasi Kelas Controller Untuk Operasi	84
Tabel 6.5 Hasil Pengujian Integrasi Kelas Controller Untuk Operasi	86
Tabel 6.6 Hasil Pengujian Integrasi Kelas Controller Untuk Operasi	87
Tabel 6.7 Kasus uji berhasil login sebagai sales	88
Tabel 6.8 Kasus uji berhasil login sebagai manager	88
Tabel 6.9 Kasus uji berhasil login sebagai admin	89
Tabel 6.10 Kasus uji gagal login	89
Tabel 6.11 Kasus uji logout	89
Tabel 6.12 Kasus uji Melihat master data motor	90
Tabel 6.13 Kasus uji membuat master data motor	90
Tabel 6.14 Kasus uji mengubah master data motor	90
Tabel 6.15 Kasus uji menghapus master data motor	91
Tabel 6.16 Kasus uji Melihat daftar harga beli	91
Tabel 6.17 Kasus uji Menambah daftar harga beli	92
Tabel 6.18 Kasus uji mengubah daftar harga beli	92
Tabel 6.19 Kasus uji menghapus daftar harga beli	93
Tabel 6.20 Kasus uji Melihat daftar harga jual	93
Tabel 6.21 Kasus uji Menambah daftar harga jual	93
Tabel 6.22 Kasus uji mengubah daftar harga jual	94
Tabel 6.23 Kasus uji menghapus daftar harga jual	94

Tabel 6.24 Kasus uji Melihat daftar permintaan barang	95
Tabel 6.25 Kasus uji membuat permintaan barang baru.....	95
Tabel 6.26 Kasus uji mengubah permintaan barang baru	95
Tabel 6.27 Kasus uji menghapus permintaan barang baru	96
Tabel 6.28 Kasus uji menghapus permintaan barang baru	96
Tabel 6.29 Kasus uji Membuat bukti penerimaan barang	97
Tabel 6.30 Kasus uji Melihat bukti penerimaan barang	97
Tabel 6.31 Kasus uji mengubah penerimaan barang baru	97
Tabel 6.32 Kasus uji menghapus bukti penerimaan barang baru.....	98
Tabel 6.33 Kasus uji Mengunci status bukti penerimaan barang	98
Tabel 6.34 Kasus uji Melihat jumlah stok barang	99
Tabel 6.35 Kasus uji Membuat order penjualan baru.....	99
Tabel 6.36 Kasus uji Melihat order penjualan	99
Tabel 6.37 Kasus uji mengubah order penjualan.....	100
Tabel 6.38 Kasus uji menghapus order penjualan	100
Tabel 6.39 Kasus uji Mencetak order penjualan.....	101
Tabel 6.40 Kasus uji Membuat surat jalan.....	101
Tabel 6.41 Kasus uji mengubah surat jalan.....	101
Tabel 6.42 Kasus uji Melihat surat jalan	102
Tabel 6.43 Kasus uji menghapus surat jalan	102
Tabel 6.44 Kasus uji Mencetak surat jalan.....	103
Tabel 6.45 Kasus uji Membuat surat faktur pajak	103
Tabel 6.46 Kasus uji mengubah surat faktur pajak	103
Tabel 6.47 Kasus uji Melihat surat faktur pajak.....	104
Tabel 6.48 Kasus uji menghapus surat faktur pajak	104
Tabel 6.49 Kasus uji Mencetak surat faktur pajak	105
Tabel 6.50 Kasus uji Membuat invoice	105
Tabel 6.51 Kasus uji mengubah invoice	105
Tabel 6.52 Kasus uji Melihat invoice	106
Tabel 6.53 Kasus uji menghapus invoice.....	106
Tabel 6.54 Kasus uji Menambah sales order pada invoice	107
Tabel 6.55 Kasus uji Mencetak detail invoice	107

Tabel 6.56 Kasus uji Membuat master data pelanggan.....	107
Tabel 6.57 Kasus uji mengubah master data pelanggan	108
Tabel 6.58 Kasus uji Melihat master data pelanggan	108
Tabel 6.59 Kasus uji menghapus master data pelanggan.....	109
Tabel 6.60 Kasus uji Membuat master data leasing	109
Tabel 6.61 Kasus uji mengubah master data leasing	110
Tabel 6.62 Kasus uji Melihat master data leasing.....	110
Tabel 6.63 Kasus uji menghapus master data leasing	110
Tabel 6.64 Kasus uji Membuat data kas master	111
Tabel 6.65 Kasus uji mengubah data kas master	111
Tabel 6.66 Kasus uji Melihat data kas master.....	112
Tabel 6.67 Kasus uji menghapus data kas master	112
Tabel 6.68 Kasus uji Membuat data kas terima	112
Tabel 6.69 Kasus uji mengubah data kas terima	113
Tabel 6.70 Kasus uji Melihat data kas terima	113
Tabel 6.71 Kasus uji menghapus data kas terima	114
Tabel 6.72 Kasus uji Membuat data kas keluar	114
Tabel 6.73 Kasus uji mengubah data kas keluar	114
Tabel 6.74 Kasus uji Melihat data kas terima	115
Tabel 6.75 Kasus uji menghapus data kas keluar.....	115
Tabel 6.76 Kasus uji Melihat riwayat kas terima	116
Tabel 6.77 Kasus uji Melihat riwayat kas terima	116
Tabel 6.78 Kasus uji Melihat riwayat sales order.....	116
Tabel 6.79 Kasus uji Menambah data antrian servis	117
Tabel 6.80 Kasus uji mengubah data antrian servis.....	117
Tabel 6.81 Kasus uji Melihat data antrian servis	118
Tabel 6.82 Kasus uji menghapus data antrian servis	118
Tabel 6.83 Kasus uji Merubah status antrian servis	118
Tabel 6.84 Kasus uji Melakukan pengecekan gratis servis	119

DAFTAR GAMBAR

Gambar 2.1 Bagan Arsitektur Laravel <i>Framework</i>	5
Gambar 2.2 <i>Waterfall Model</i>	7
Gambar 2.3 Flowchat dan Flow graph	12
Gambar 3.1 Diagram Alir Metodologi	14
Gambar 4.2 <i>Use Case Diagram</i> Sistem Eksternal	31
Gambar 4.1 <i>Use Case Diagram</i> Sistem Utama	32
Gambar 5.1 <i>Sequence Diagram</i> Membuat order penjualan baru	57
Gambar 5.2 <i>Sequence Diagram</i> Membuat bukti penerimaan barang	58
Gambar 5.3 <i>Sequence Diagram</i> Lock Invoice	58
Gambar 5.4 <i>Class Diagram</i> pada Controller	59
Gambar 5.5 <i>Class Diagram</i> pada Model	60
Gambar 5.7 Perancangan <i>Physical Data Model</i> (PDM)	62
Gambar 5.6 Perancangan ERD	63
Gambar 5.8 Perancangan antarmuka sales order	66
Gambar 5.9 Perancangan antarmuka list invoice	67
Gambar 5.10 Perancangan antarmuka master pelanggan	68
Gambar 5.11 Perancangan antarmuka faktur penjualan	69
Gambar 5.12 Perancangan antarmuka kas master	70
Gambar 5.13 Perancangan antarmuka penerimaan barang	71
Gambar 5.14 Implementasi Database	74
Gambar 5.15 Implementasi antarmuka list sales order	75
Gambar 5.16 Implementasi antarmuka invoice	75
Gambar 5.17 Implementasi antarmuka master pelanggan	76
Gambar 5.18 Implementasi antarmuka faktur penjualan	76
Gambar 5.19 Implementasi antarmuka kas master	77
Gambar 5.20 Implementasi antarmuka penerimaan barang	77

DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	124
---------------------------------	-----



BAB 1 PENDAHULUAN

1.1 Latar belakang

Dengan semakin berkembangnya perekonomian yang ditandai dengan adanya era globalisasi dan kemajuan teknologi yang pesat memungkinkan tersedianya sarana bagi pemenuhan kebutuhan pelanggan akan barang dan jasa menjadi lebih mudah. Situasi tersebut akan menimbulkan persaingan yang ketat antar perusahaan baik yang menghasilkan produk maupun jasa. Dalam era globalisasi saat ini dunia bisnis telah dipenuhi oleh banyak perusahaan yang jenisnya juga bermacam-macam. Salah contoh adalah perusahaan bidang motor. Dalam sebuah perusahaan bidang motor harus dapat mengelola barang motor hingga sampai pada tangan pelanggan dengan baik. *After sales* yang dilakukan perusahaan juga penting terhadap pelanggan, hal ini mampu meningkatkan penjualan dan kepuasan pelanggan.

Hal tersebut berlaku juga bagi perusahaan yang bergerak di bidang transportasi dalam hal ini penjualan sepeda motor yaitu PT. Mega Abadi Motorindo. PT. Mega Abadi Motorindo merupakan salah satu cabang perusahaan PT. Kawasaki Motor Indonesia (KMI) di Indonesia. Perusahaan ini merupakan dealer motor retail yang menjual produk Kawasaki di daerah Kota Malang, baik dari motor trail hingga motor gede atau biasa disebut moge.

Sebagai pendukung dalam berjalannya sebuah dealer motor PT. Mega Abadi Motorindo adalah adanya stok barang yang terdapat pada dealer tersebut. Hal tersebut menjadi kebutuhan utama bagi dealer tersebut. Untuk mendapatkan stok barang, PT. Mega Abadi Motorindo harus melakukan permintaan barang kepada gudang pusat. Setelah barang telah sampai, sales akan melakukan pengecekan apakah barang yang datang sudah dikirim dengan benar. Selama ini proses tersebut dilakukan dengan menggunakan email sebagai media komunikasi dengan gudang pusat dan *Microsoft Excel* sebagai aplikasi untuk mengolah data, baik dari penerimaan barang, penjualan hingga mencetak surat jalan. Hal ini tidak efisien dari sisi sales, karena sales harus melakukan pengecekan data yang dikirim menggunakan email dan data yang ada di *Microsoft Excel*. *Manager* dari PT. Mega Abadi Motorindo juga mengeluhkan sulitnya untuk melihat riwayat penjualan jika mendadak, karena manager harus melihat data yang ada pada komputer sales.

Setelah penjualan barang selesai dilakukan, hubungan dengan pelanggan harus tetap dijaga. Pelanggan bisa mengetahui info tentang promo, baik promo pembelian sepeda maupun pengingat waktu servis motor langsung melalui layanan pesan sosial, seperti *Whatsapp* maupun *Line Messenger*. Hal ini perlu dilakukan oleh perusahaan, karena menurut Kalakota dan Robinson (2001) melakukan hubungan dengan pelanggan akan meningkatkan pendapatan pada perusahaan dan meminimalisir kekecewaan dari pelanggan.

Solusi yang akan diambil oleh penulis untuk menyelesaikan permasalahan diatas adalah membuat sistem yang mampu menangani permintaan dan penerimaan barang, penjualan barang, serta layanan pelanggan yang ada pada PT. Mega Abadi Motorindo.

Pada proses implementasi kode program dari sistem yang dirancang penulis menggunakan sebuah framework yang menggunakan bahasa pemrograman PHP. Salah satu framework PHP yang cukup banyak digunakan adalah Laravel, karena Laravel memiliki beberapa kelebihan yaitu update security secara berkala dari pengembang.

Berdasarkan pemaparan diatas, penulis mengambil judul **“Pembangunan Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo”** dengan harapan perangkat lunak yang akan dibangun dapat menyelesaikan permasalahan yang sedang dihadapi oleh PT. Mega Abadi Motorindo.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, dapat dirumuskan permasalahan dalam pembuatan skripsi, diantaranya:

1. Apa saja kebutuhan dalam sistem Pembangunan Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo?
2. Bagaimana perancangan dan implementasi Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya?
3. Bagaimana pengujian pada Aplikasi Penjualan dan Pelayanan Pelanggan?

1.3 Tujuan

Tujuan dari pembangunan sistem penjualan dan pelayanan kepada pelanggan adalah untuk membuat suatu sistem penjualan dan pelayanan yang dapat mempermudah karyawan PT. Mega Abadi Motorindo

1.4 Manfaat

Berdasarkan penjelasan yang dipaparkan pada latar belakang, manfaat yang akan diperoleh dari penelitian adalah bagi instansi dapat membantu mempermudah proses penjualan dan pelayanan pelanggan dan bagi pembaca sebagai bahan acuan dalam merancang perangkat lunak yang memiliki keterkaitan dengan topik ini

1.5 Batasan masalah

1. Sistem dirancang khusus untuk PT. Mega Abadi Motorindo.
2. Sistem hanya mampu untuk menangani penjualan barang dan pelayanan pelanggan

1.6 Sistematika pembahasan

Untuk mempermudah pemahaman laporan skripsi ini, penulis membuat sistematika penulisan yang mengemukakan secara singkat mengenai isi tiap-tiap bab.

BAB I Pendahuluan

Bagian pendahuluan dari laporan yang memberikan gambaran seperti latar belakang, rumusan masalah, tujuan, serta manfaat , batasan masalah, dan sistematika penulisan.

BAB II Landasan Kepustakaan

Pada bab ini berisi tentang dasar teori dan referensi yang mendasari proses perancangan sistem.

BAB III Metodologi Penelitian

Pada bab ini berisi tentang metode dan langkah kerja yang dilakukan pada penelitian.

BAB IV Analisis Kebutuhan Sistem

Pada analisis perancangan yang akan dilakukan adalah membahas dan melakukan analisis dari penelitian dan perancangan penelitian yang dilakukan. Hal tersebut dilakukan agar mudah untuk diambil kesimpulan dari analisis dan perancangan suatu penelitian.

BAB V Perancangan dan Implementasi

Menjelaskan proses implementasi sistem dari analisis kebutuhan dan perancangan.

BAB VI Pengujian

Menjabarkan mengenai proses pengujian, hasil pengujian, serta kesulitan yang dihadapi selama tahap pengujian sistem.

BAB VII Kesimpulan dan Saran

Penutup dari sebuah skripsi berisi tentang kesimpulan yang diperoleh dari hasil penelitian serta menjawab apa yang telah dirumuskan dalam rumusan masalah. Serta saran yang dilakukan untuk pengembangan dari penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tinjauan pustaka ini digunakan membandingkan penelitian yang relevan dengan beberapa penelitian sebelumnya. Terdapat 2 penelitian yang berhubungan dengan penjualan dan pelayanan pelanggan, yakni penelitian mengenai Analisis Sistem Informasi Penjualan dan Persediaan pada PT. XYZ dan penelitian mengenai Analisis Sistem Informasi Penjualan dan Persediaan pada PT. XYZ.

Dalam penelitian mengenai Analisis Sistem Informasi Penjualan dan Persediaan pada PT. XYZ yang dilakukan oleh Hanny Juwitasari, Marlene Martani, dan Arya Nata Gani Putra (Juwitasari, Martani, Putra, 2015) mereka melakukan pengembangan sistem menggunakan pendekatan *object oriented* dalam melakukan analisis, perancangan, dan implementasi. Dimana sistem tersebut memiliki tiga aktor, yakni *logistic, purchasing, dan accounting*. Fitur-fitur yang tersedia yakni membuat formulir dan laporan pemasukan, penjualan, dan persediaan barang. Mengelola data pemasukan dan pengeluaran barang dengan menggunakan *Microsoft visual studio 2010, crystal report* untuk menghasilkan laporan bagi perusahaan, dan basis data menggunakan SQL Server 2005.

Lalu dalam penelitian mengenai Analisis Sistem Informasi Penjualan dan Persediaan pada PT. XYZ yang dilakukan oleh Devyano Luhukay, Yohannes Kurniawan dan Titan (Luhukay, Kurniawan, Titan, 2013) mereka melakukan pengembangan sistem menggunakan pendekatan *object oriented*. Dimana sistem memiliki 3 aktor, yakni marketing, kasir dan bagian gudang. Fitur-fitur yang tersedia yakni membuat formulir persediaan daging sapi, laporan pemasukan, penjualan dan persediaan daging sapi, mendaftarkan *customer*, mengecek ketersediaan daging sapi dan mencatat payment. Sistem dibangun berbasis desktop dengan menggunakan bahasa C#.

2.2 Bahasa Pemrograman PHP

2.2.1 Definisi PHP

PHP merupakan sebuah bahasa pemrograman yang dipasang pada dokumen HTML. Sebagian besar sintaks pada PHP mempunyai kemiripan dengan bahasa C, Java dan Perl, namun pada PHP ada beberapa fungsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan PHP adalah untuk memungkinkan perancang web yang dinamis. (Ratna:2014)

2.3 Laravel Framework

2.3.1 Definisi Framework

Framework dapat diartikan sebagai kumpulan potongan-potongan program yang diusulkan atau diorganisasikan sedemikian rupa, sehingga dapat digunakan

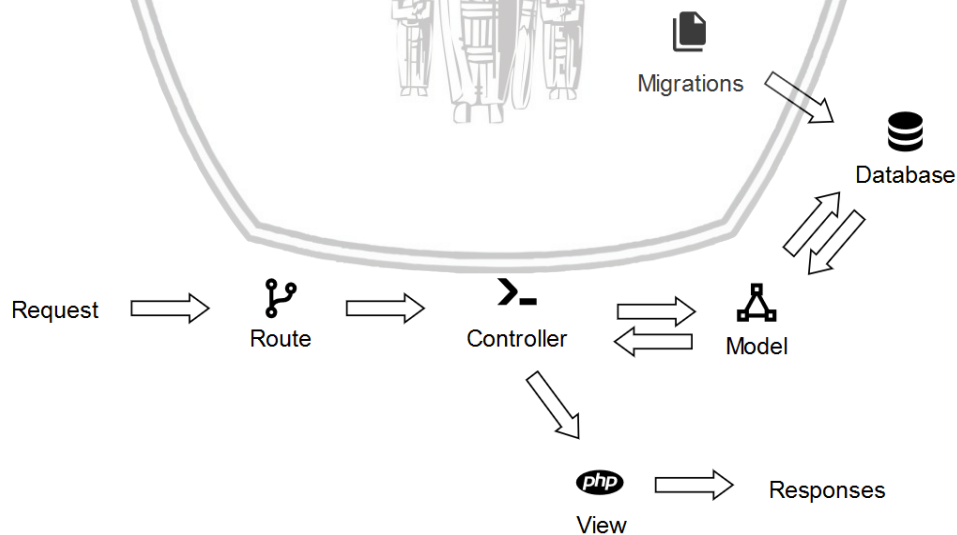
untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal. Dengan menggunakan *framework*, sebuah aplikasi akan tersusun dengan rapi. Namun, menggunakan *framework* bukan berarti kita bebas dari pengkodean. Programmer diharuskan menggunakan variabel dan fungsi-fungsi yang ada di sebuah *framework* itu. Karena itulah, kerja programmer bisa menjadi efektif karena tidak harus membuat fungsi-fungsi lagi dari awal.

2.3.2 Definisi Laravel Framework

Laravel merupakan sebuah *framework* atau disebut kerangka kerja yang digunakan untuk membangun sebuah sistem yang berbasis PHP. Laravel termasuk kerangka kerja yang *lightweight* (ringan) untuk mendukung dalam pengembangan aplikasi siap pakai.

Gambar 2.1 adalah arsitektur dari sistem *Laravel Framework*, yang dibagi dari 6 modul utama, yaitu *route*, *controller*, *model*, *view*, *database* dan *migrations*.

Route merupakan modul yang berfungsi untuk mengatur URL yang dapat diakses oleh user. *Route* menerima request yang berasal dari user untuk diteruskan kepada *controller* yang dituju. *Controller* merupakan tempat dimana segala kebutuhan laravel dieksekusi. *Controller* dapat berkomunikasi dengan model apabila diperlukan data dari database dan menampilkannya melalui view. Model merupakan representasi dari database itu sendiri. Ini merupakan implementasi dari object-relational mapping. Pada laravel ini disebut dengan *Eloquent*. Database yang didukung oleh laravel adalah MySQL, Postgres, SQLite dan SQL Server. *Migrations* dibutuhkan untuk melakukan pembuatan schema database. *View* untuk menampilkan data yang akan ditujukan pada user.



Gambar 2.1 Bagan Arsitektur Laravel Framework

(Sumber: Laravel Documentation)

2.4 MySQL

2.4.1 Definisi MySQL

MySQL adalah database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi, yaitu free software dan shareware. MySQL pertama kali dirintis oleh Michael Widenius. (Wahana Komputer, 2010)

2.4.2 Fitur dan Kelebihan MySQL

Kelebihan MySQL dibandingkan dengan sistem manajemen database yang lain menurut Wahana Komputer (2010) adalah

1. MySQL mempunyai performa yang tinggi namun simple
2. Database MySQL mengerti bahasa SQL
3. MySQL dapat diakses melalui protocol ODBC (*Open Database Connectivity*) buatan Microsoft.
4. Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu
5. MySQL merupakan database yang mampu menyimpan data berkapasitas besar.

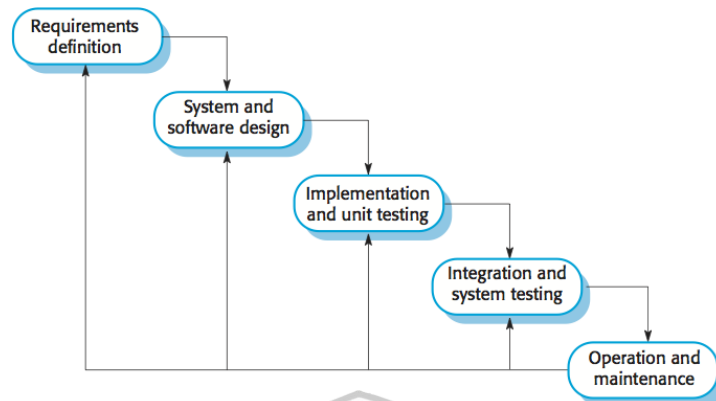
2.5 Software Development Life Cycle

Software Development Life Cycle atau disingkat SDLC adalah kerangka rekayasa perangkat lunak yang tepat dan penting untuk mengembangkan aplikasi yang handal, dapat diperlihara, dan hemat biaya. (S. K. Singh, 2009). SDLC merupakan sebuah cara yang digunakan dalam perancangan pengembangan, dan pengujian perangkat lunak. Tujuan dari SDLC adalah untuk menghasilkan perangkat lunak dengan kualitas terbaik yang memenuhi kebutuhan user, dan selesai dalam waktu yang ditentukan dengan biaya yang sesuai.

2.5.1 SDLC Waterfall

Untuk memilih proses pengembangan perangkat lunak selalu tergantung pada hasil analisis masalah yang dihadapi. Pada penelitian ini digunakan SDLC waterfall dalam membangun sistem. Pemilihan model waterfall dianggap sesuai karena pelaksanaannya dilakukan secara bertahap, sehingga menghasilkan kualitas sistem yang baik. Pengembangan dengan menggunakan waterfall dilakukan dengan menyelesaikan satu fase, lalu melanjutkan ke fase berikutnya. Model *waterfall* sendiri terdiri dari proses *requirement definition, system and software*

design, implementation and unit testing, intergration and system testing, operation and maintenance seperti yang digambarkan pada Gambar 2.2



Gambar 2.2 Waterfall Model

Sumber : (Sommerville, 2011)

1. *Requirement definition*

Pada proses ini dilakukan analisis dan pendefinisian kebutuhan sistem, batasan, tujuan hingga ditetapkan menjadi spesifikasi sistem yang telah dikonsultasikan dengan pengguna sistem.

2. *System and software design*

Pada proses ini dilakukan perancangan sistem dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan proses identifikasi dan mendeskripsikan abstraksi sistem perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

Pada proses ini, rancangan perangkat lunak diwujudkan dalam bentuk unit. Unit testing ini dilakukan untuk memverifikasi setiap unit agar sesuai dengan spesifikasinya.

4. *Integration and system testing*

Pada proses ini setiap unit program diintegrasikan dan diuji sebagai sistem yang lengkap untuk memastikan bahwa kebutuhan perangkat lunak telah dipenuhi. Setelah diuji perangkat lunak diserahkan kepada pengguna

5. *Operation and maintenance*

Pada proses ini sistem dipasang dan dikondisikan pada penggunaan sebenarnya. Maintenance melibatkan proses memperbaiki error yang mana tidak ditemukan pada tahap awal pengembangan, meningkatkan implementasi unit sistem dan meningkatkan layanan sistem ketika ditemukan kebutuhan baru.

2.6 UML

Unified Modeling Language (UML) adalah salah satu perkakas yang digunakan untuk melakukan analisis dan perancangan sistem (Adi Nugroho). UML merupakan bahasa visual yang menjadi standar untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak (Sugiarti, 2013: 34).

2.6.1 Use Case Diagram




Use case diagram menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Sugiarti, 2013: 41).

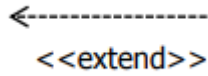
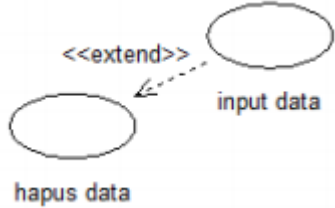
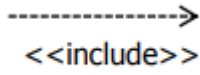
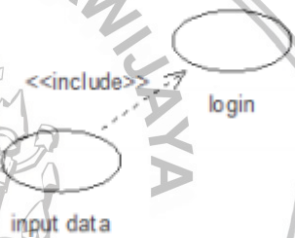
Penamaan pada *use case* didefinisikan sesederhana mungkin dan mudah untuk dipahami. Ada dua hal utama dalam *use case*, yaitu aktor dan *use case*.

1. Aktor : merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Meskipun simbol dari aktor berbentuk orang, tapi aktor belum tentu merupakan orang.
2. Use case : merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Menurut Sugiarti (2013: 42), dalam *use case diagram* terdapat beberapa simbol yang digunakan sebagai berikut :

Tabel 2.1 Simbol simbol pada Use Case Diagram


Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan oleh sistem sebagai unit.
<p><i>Aktor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dikembangkan di luar sistem tersebut.
<p><i>Asosiasi</i></p> 	Menunjukkan hubungan antara aktor dan use case atau antar use case.


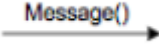



<p><i>Extend</i></p> 	<p>Hubungan use case tambahan ke use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; anak panah menuju use case yang dituju, contoh :</p> 
<p><i>Include</i></p> 	<p>Hubungan antara use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya, sehingga use case harus dijalankan terlebih dulu sebelum menjalankan use case tambahan; arah panah menunjuk pada use case yang perlu dijalankan terlebih dahulu, contoh :</p> 

2.6.2 Sequence Diagram

Diagram sekuen merupakan diagram yang digunakan untuk menggambarkan perilaku objek pada *use case* dengan mendeskripsikan proses objek dengan pesan yang dikirimkan. Oleh karena itu, untuk menggambar diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Sugiarti, 2013: 69).

Tabel 2.2 Simbol simbol pada *sequence diagram*

Simbol	Deskripsi
<p><i>Lifeline</i></p> 	Antarmuka yang saling berinteraksi/


Actor 	Digunakan untuk menggambarkan user atau pengguna
Message 	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
Boundary 	Menggambarkan sebuah form
Control Class 	Digunakan untuk menghubungkan boundary dengan tabel
Entity Class 	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan







2.6.3 Class Diagram

Class diagram digunakan untuk menggambarkan struktur dalam objek sistem. Diagram ini menunjukkan class object yang menyusun sistem dan juga hubungan antara class object (Sugiarti, 2013: 37).

Kelas memiliki atribut dan metode atau operasi. Atribut adalah variabel variabel yang mendeskripsikan properti dengan bentuk sebaris teks dalam kelas tersebut, sedangkan metode adalah fungsi yang dimiliki oleh kelas yang dalam *class diagram* dilambangkan menggunakan simbol-simbol. (Sugiarti, 2013: 57-59).

Tabel 2.3 Simbol simbol pada *class diagram*

Simbol	Deskripsi
Package 	Sebuah bungkusan dari satu atau lebih kelas.
Kelas	Tiap kelas memiliki nama, attribute, dan operation atau method.

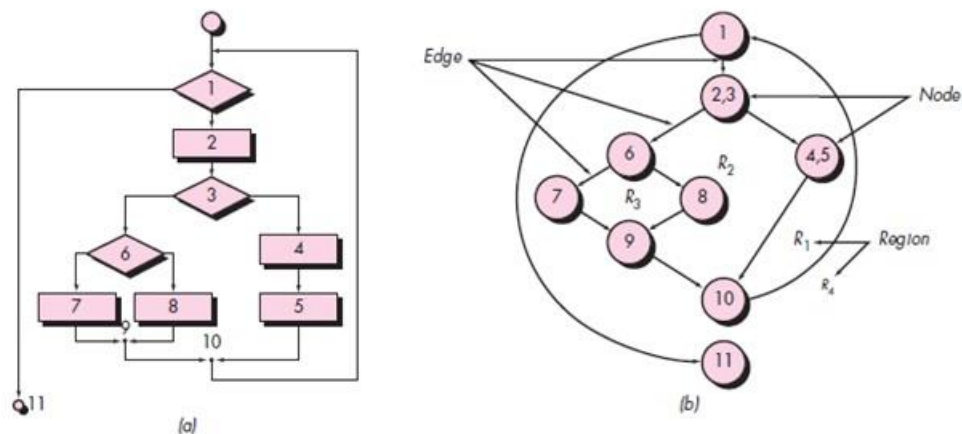
<div> <div>nama kelas</div> <div>+Attribute1</div> <div>+Attribute2</div> <div>+Operation1()</div> <div>+Operation2()</div> </div>		
<div>Interface</div> 		Sama seperti konsep interface dalam pemrograman berorientasi objek.
<div>Asosiasi</div> 		Hubungan antar kelas dengan pengertian umum.
<div>Asosiasi berarah</div> 		Hubungan antar kelas dengan pengertian kelas yang satu digunakan oleh kelas yang lain.
<div>Generalisasi</div> 		Hubungan antar kelas dengan pengertian generalisasi spesialisasi.
<div>Kebergantungan</div> 		Hubungan antar kelas dengan pengertian kebergantungan antar kelas.
<div>Agregasi</div> 		Hubungan antar kelas dengan makna semua-sebagian.

2.7 Pengujian

Proses akhir yang dilakukan dalam pengembangan perangkat lunak adalah melakukan pengujian pada sistem yang selesai dibangun. Pengujian perangkat lunak merupakan hal yang sangat penting karena merupakan sebuah aktifitas untuk melakukan pemeriksaan terhadap error yang mungkin terjadi karena kesalahan dari perangkat lunak (Pressman, 2010). Dalam penelitian ini, pengujian yang digunakan adalah *whitebox testing* dan *blackbox testing*.

2.7.1 Whitebox testing

Menurut Pressman, *whitebox testing* atau dapat disebut juga sebagai *glass-box testing* merupakan cara pengujian dengan melibatkan algoritme pengembangan sebagai bahan yang diujikan (*test case*). Dengan menggunakan *whitebox testing*, developer perangkat lunak dapat memberikan sebuah jaminan bahwa seluruh jalur dapat ditelusuri serta terdapat batas berhenti *pada looping*. *Basis Path Testing* merupakan salah satu *whitebox testing* yang digunakan pada penelitian ini. *Basis Path testing* dimulai dengan menentukan algoritme yang selanjutnya akan dibuat menjadi notasi *flow graph* (Pressman, 2010).



Gambar 2.3 Flowchat dan Flow graph

Sumber : Pressman (2010)

Setelah didapatkan flowgraph maka selanjutnya menentukan jalur atau yang biasa disebut independent path dari node-node serta region yang terbentuk. Sebagai contoh jalur atau independent path yang terbentuk dari gambar 2.3b adalah sebagai berikut

Path 1 = 1 - 11

Path 2 = 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

Path 3 = 1 - 2 - 3 - 6 - 8 - 9 - 10 - 1 - 11

Path 4 = 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

Independent path telah terbentuk maka dapat dilakukan perhitungan dalam mencari jumlah jalur atau path dalam sebuah flow graph. Untuk melakukan perhitungan tersebut dikenal dengan nama cyclomatic complexity. *Cyclomatic complexity* merupakan sebuah rumusan perhitungan yang dapat digunakan untuk mengukur tingkatan kompleksitas dari sebuah algoritme (Pressman, 2010). Rumus cyclomatic complexity yang dijelaskan oleh Pressman dalam buku software engineering dapat dilihat pada persamaan A1, persamaan A2, dan persamaan A3 berikut ini :

$$V(G) = E - N + 2 \dots\dots\dots (A1)$$

$$V(G) = P + 1 \dots\dots\dots (A2)$$

$$V(G) = \text{Jumlah region} \dots\dots\dots (A3)$$

Dimana :

E = jumlah edge

N = jumlah node

P = jumlah predictive node

Semakin banyak jalur independent yang ditemukan dalam algoritme yang diujikan, maka akan semakin sulit sistem tersebut untuk diujikan dan memiliki cost yang

tinggi pula seperti pada tabel 2.2 berikut yang meruokan tabel angka dari cyclomatic complexity.

Tabel 2.4 Tingkatan angka cyclomatic complexity

Cyclomatic Complexity	Level Kompleksitas	Dapat diuji	Cost dan effort
1-10	Rendah	Tinggi	Rendah
10-20	Tinggi	Sedang	Sedang
20-40	Sangat tinggi	Rendah	Tinggi
>40	-	Tidak dapat diuji	Sangat tinggi

Sumber : Madhavi, 2016

2.7.2 Blackbox testing

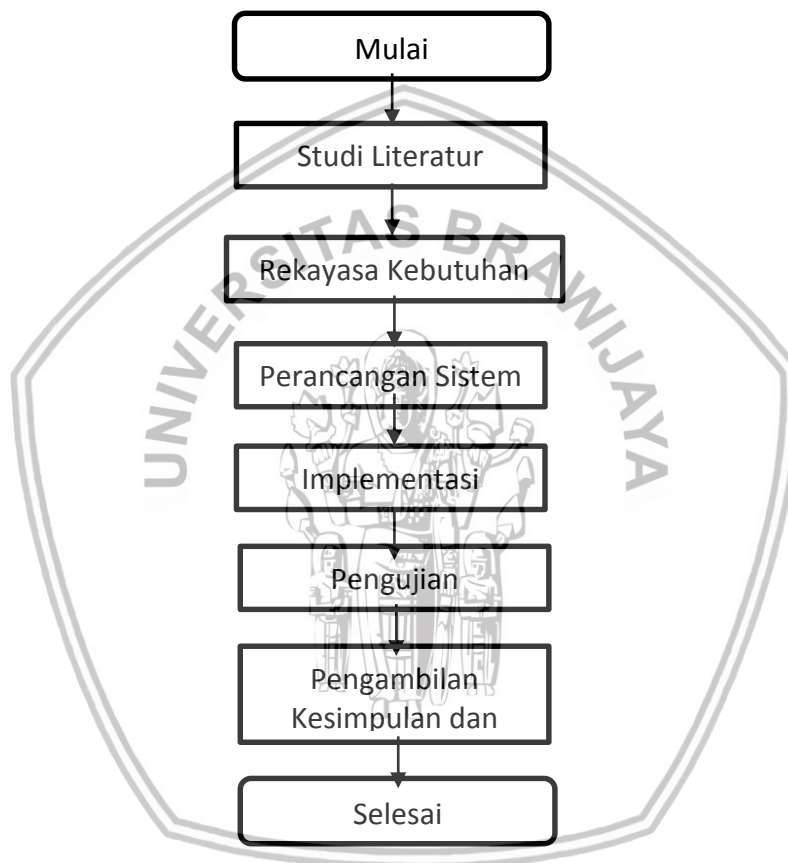
Menurut Pressman, *Blackbox testing* atau yang disebut juga sebagai *behavioral testing*, pengujian ini merupakan pengujian yang berfokus kepada fungsional sistem. *Blackbox testing* melibatkan sejumlah fungsi pada sistem sebagai bahan uji, setiap fungsi diuji dengan scenario yang telah dibuat. *Blackbox testing* dilakukan sebagai pengujian untuk percobaan pencarian error pada sistem dengan kategori (Pressman, 2010) :

1. Terjadinya kesalahan pada struktur data dan data yang diakses dalam database
2. Adanya error pada antarmuka
3. Fungsional sistem yang hilang serta tidak benar

Blackbox testing dilakukan sebagai pelengkap dari whitebox testing dan buka sebagai pengganti. Untuk melakukan pengujian ini, pengembang perangkat lunak membuat scenario yang terbagi dalam kasus uji yang benar serta kasus uji yang salah dan dari kasus tersebut dapat dilihat feedback yang diberikan oleh sistem apakah telah sesuai dengan kebutuhan yang diharapkan atau belum.

BAB 3 METODOLOGI PENELITIAN

Metode penelitian menjelaskan metode yang digunakan dalam pembangunan sistem pada PT. Mega Abadi Motorindo. Pada penelitian ini menggunakan metode penelitian waterfall. Dalam pelaksanaan penelitian ini meliputi langkah-langkah yang akan dilakukan dalam pengerjaan tugas akhir, yaitu studi literatur, analisis kebutuhan, perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Berikut adalah diagram alir dari metodologi.



Gambar 3.1 Diagram Alir Metodologi

3.1 Studi Literatur

Studi literatur digunakan untuk mencari dasar – dasar teori yang mendukung serta menunjang dari penelitian. Penelusuran yang mendukung penelitian di dapat dari jurnal, web, buku dan skripsi serta laporan – laporan penelitian. Studi teori yang dapat mendukung antara lain:

1. Rekayasa Perangkat Lunak

Sebagai dasar teori dalam melakukan menentukan tahapan tahapan pengembangan sistem

2. *Unified Modeling Language*

Sebagai dasar teori dalam melakukan pemodelan objek pada tahapan perancangan perangkat lunak

3. *Entity Relationship Diagram*

Sebagai dasar teori dalam melakukan perancangan database pada tahapan perancangan perangkat lunak

4. *Laravel Framework*

Sebagai dasar teori lingkungan aplikasi

3.2 Rekayasa Kebutuhan

Tahapan analisis kebutuhan merupakan langkah pertama dalam melakukan penelitian yang harus dilakukan, agar hasil analisis kebutuhan dapat digunakan dengan baik dalam perancangan sistem. Berikut merupakan penjelasan tahapan dalam analisis kebutuhan yang dilakukan :

1. Menganalisis kebutuhan menjadi kebutuhan dan spesifikasi kebutuhan yang masing-masing telah diidentifikasi dengan aturan penomoran
2. Hasil dari analisis kebutuhan dikategorikan dalam beberapa kebutuhan fungsional dan nonfungsional dan mengidentifikasi aktor
3. Melakukan pemodelan kebutuhan *use case diagram* dari setiap kebutuhan yang terdapat pada kebutuhan fungsional.
4. Setiap *use case* yang terdapat pada *use case diagram* dijelaskan lebih detail pada *Use Case Scenario*

Hasil pemodelan kebutuhan *Use Case Diagram* dan *Use Case Scenario* dari hasil pemetaan kebutuhan fungsional akan digunakan dalam melakukan perancangan sistem selanjutnya.

3.3 Perancangan Sistem

Tahap perancangan sistem dilakukan untuk mempermudah dalam implementasi. Hasil dari tahap analisis kebutuhan digunakan sebagai acuan dalam melakukan tahap perancangan, yang dilakukan menggunakan pendekatan *object oriented*. Berikut merupakan penjelasan tahap dalam perancangan sistem yang dilakukan :

1. Melakukan pemodelan *sequence diagram* dari hasil pemodelan kebutuhan *Use Case Scenario*. Hanya akan ada 5 sampel *sequence diagram* yang dijelaskan.
2. Hasil dari masing-masing *sequence diagram* dikelompokkan menjadi beberapa kelas pada pemodelan *class diagram*

3. Membuat algoritme masing masing fungsi di setiap klas, hasil dari pemodelan *class diagram* sebagai acuan dalam tahap implementasi. Hanya ada nada 3 sampel algoritme yang dijelaskan
4. Membuat model *entity relationship diagram* dalam melakukan perancangan basis data
5. Membuat rancangan antarmuka yang akan diimplementasikan selanjutnya. Hanya ada beberapa rancangan antarmuka, yang merupakan bagian penting dari sistem yang akan dijadikan sampel.

Hasil pemodelan class diagram beserta *algoritme* yang dijelaskan masing-masing fungsi dan hasil perancangan basis data secara *physical* akan digunakan pada saat melakukan implementasi sistem dan rancangan antar muka akan digunakan pada saat melakukan implementasi tampilan sistem.

3.4 Implementasi Sistem

Implementasi sistem adalah fase pembangunan sistem yang mengacu pada perancangan sistem. Sistem akan diimplementasikan dengan menggunakan framework Laravel sebagai sistem. Proses yang ada pada tahap implementasi ini adalah sebagai berikut

1. Spesifikasi Sistem

Spesifikasi sistem yang akan digunakan untuk membangun sistem yang akan dibuat meliputi perangkat keras, perangkat lunak dan sistem operasi

2. Implementasi Basis Data

Implementasi basis data mengacu kepada basis data bagian perancangan sistem. Implementasi basis data menggunakan DBMS (*Database Management System*) MySQL.

3. Implementasi Antarmuka

Implementasi antarmuka mengacu kepada perancangan antarmuka yang ada pada bagian perancangan sistem. Implementasi antarmuka ini menggunakan bahasa pemrograman HTML, CSS, dan javascript.

3.5 Pengujian Sistem

Pengujian sistem merupakan proses untuk mengetahui apakah sistem yang dibangun telah memenuhi dan berjalan sesuai dengan kebutuhan yang telah didefinisikan. Pengujian juga berfungsi untuk meminimalisir atau menghilangkan masalah yang masih mungkin ada didalam sistem yang telah dibuat.

Pengujian yang dilakukan adalah pengujian dari white box atau menguji kode program dengan metode seperti pengujian unit dan pengujian integrasi dan. Sedangkan dalam pengujian blackbox dapat digunakan cara pengujian validasi. Pengujian tersebut dilakukan dengan kasus uji tertentu sesuai fungsi program sehingga didapatkan hasil yang sesuai dengan perkiraan dari penguji aplikasi.

3.6 Kesimpulan dan Saran

Kesimpulan merupakan hasil akhir yang menjawab permasalahan dari rumusan masalah. Kesimpulan didapat dengan cara membuat sistem dan melakukan penelitian. Lalu melakukan pengujian dari sistem dan data yang diperoleh. Setelah hasil dari pengujian tersebut sudah sesuai maka dilakukan analisis dari kesesuaian masalah yang telah terdefinisi sebelumnya. Dari langkah – langkah tersebut maka dapat disimpulkan dengan singkat dari proses, hasil serta, kelebihan dari proyek yang sedang dibuat dengan melihat dampak yang dapat terjadi dalam penggunaan aplikasi tersebut.



BAB 4 ANALISIS KEBUTUHAN SISTEM

Bab ini berisi analisis kebutuhan dan perancangan dalam pembangunan sistem aplikasi terintegrasi pada PT. Mega Abadi Motorindo. Pada tahap ini terdapat beberapa tahap, yaitu gambaran umum sistem, identifikasi aktor, spesifikasi kebutuhan, pemodelan kebutuhan, dan perancangan sistem. Dari tahap tersebut akan dijelaskan bagaimana sistem ini akan berjalan, mendeskripsikan aktor serta apa saja yang dapat dilakukan, pengumpulan data serta analisis data apa saja yang diperlukan oleh pengguna, dan identifikasi kebutuhan pengguna baik itu kebutuhan fungsional maupun kebutuhan non fungsional.

4.1 Gambaran Umum Sistem

Berdasarkan permasalahan yang ada pada latar belakang, sistem ini harus bisa digunakan oleh admin, sales counter, kasir, mekanik serta manager pemilik perusahaan PT Mega Abadi Motorindo. Admin dapat melakukan manajemen master data, baik dari penambahan, pembaharuan dari model motor hingga daftar harga motor yang dijual kepada pelanggan. Sales counter dapat melakukan permintaan barang kepada gudang pusat, hingga melihat history atau jurnal transaksi yang pernah terjadi. Sales juga dapat mengirimkan informasi baik ke masing masing pelanggan maupun semua pelanggan terkait info, promo dan kebutuhan lain yang mendukung pelayanan terhadap pelanggan. Kasir melakukan manajemen data kas pada PT. Mega Abadi Motorindo. Mekanik dapat melakukan registrasi antrian servis motor pada pelanggan yang akan melakukan servis, mekanik disini juga dapat melihat riwayat apakah motor yang dimiliki pelanggan masih memiliki layanan gratis servis.

4.2 Analisis Kebutuhan

Analisis kebutuhan merupakan proses penentuan kebutuhan sistem. Proses ini diawali dengan elisitasi kebutuhan, identifikasi aktor, spesifikasi kebutuhan dan pembuatan model kebutuhan dengan menggunakan *use case diagram* dan *use case scenario*.

4.2.1 Elisitasi Kebutuhan

Elisitasi kebutuhan adalah proses pertama dalam analisis kebutuhan. Elisitasi kebutuhan dilakukan untuk memahami masalah yang terjadi pada PT. Mega Abadi Motorindo. Kemudian, dari permasalahan yang didapatkan akan diajukan solusi-solusi yang dapat membantu menyelesaikan permasalahan yang terjadi pada PT. Mega Abadi Motorindo.

Teknik elisitasi yang digunakan dalam penelitian ini adalah wawancara. Proses wawancara dilakukan untuk mengetahui permasalahan yang terjadi, orang-orang yang terlibat, dan proses bisnis yang terjadi pada PT. Mega Abadi Motorindo yang hasilnya akan didapatkan fitur-fitur dan aktor-aktor yang terlibat dalam sistem.

Berdasarkan hasil wawancara didapatkan kebutuhan dalam sistem penjualan dan pelayanan pada PT. Mega Abadi Motorindo yaitu :

1. Permintaan Barang
 - 1.1 Pada bagian permintaan barang harus memiliki fungsi untuk melakukan permintaan barang kepada gudang pusat
 - 1.2 Pada bagian permintaan barang harus memiliki fungsi untuk melihat daftar permintaan barang
2. Penerimaan Barang
 - 2.1 Pada bagian penerimaan barang harus memiliki fungsi untuk menambah barang
 - 2.2 Pada bagian penerimaan barang harus memiliki fungsi untuk merubah daftar penerimaan barang
 - 2.3 Pada bagian penerimaan barang harus memiliki fungsi untuk melihat daftar penerimaan barang
3. Sales Order
 - 3.1 Pada bagian sales order harus memiliki fungsi untuk menambah sales order baru
 - 3.2 Pada bagian sales order harus memiliki fungsi untuk merubah sales order
 - 3.3 Pada bagian sales order harus memiliki fungsi untuk melihat daftar sales order
4. Delivery Order
 - 4.1 Pada bagian delivery order harus memiliki fungsi untuk menambah delivery order
 - 4.2 Pada bagian delivery order harus memiliki fungsi untuk merubah delivery order
 - 4.3 Pada bagian delivery order harus memiliki fungsi untuk melihat daftar delivery order
5. Faktur Penjualan
 - 5.1 Pada bagian faktur penjualan harus memiliki fungsi untuk menambah faktur penjualan
 - 5.2 Pada bagian faktur penjualan harus memiliki fungsi untuk merubah faktur penjualan
 - 5.3 Pada bagian penerimaan barang harus memiliki fungsi untuk melihat daftar faktur penjualan
6. Invoice
 - 6.1 Pada bagian invoice harus memiliki fungsi untuk menambah invoice

- 6.2 Pada bagian invoice harus memiliki fungsi untuk merubah invoice
- 6.3 Pada bagian invoice harus memiliki fungsi untuk melihat daftar invoice
7. Biodata Pelanggan
 - 7.1 Pada bagian biodata pelanggan harus memiliki fungsi untuk menambah data pelanggan
 - 7.2 Pada bagian biodata pelanggan harus memiliki fungsi untuk merubah data pelanggan
 - 7.3 Pada bagian biodata pelanggan harus memiliki fungsi untuk melihat daftar biodata pelanggan
8. Layanan Pelanggan
 - 8.1 Pada bagian layanan pelanggan harus memiliki fungsi untuk mengirimkan pesan whatsapp ke salah satu pelanggan
 - 8.2 Pada bagian layanan pelanggan harus memiliki fungsi untuk mengirimkan pesan whatsapp ke semua pelanggan
 - 8.3 Pada bagian layanan pelanggan harus memiliki fungsi untuk melihat daftar pesan whatsapp yang telah dikirimkan
9. Master Model
 - 9.1 Pada bagian master model harus memiliki fungsi untuk menambah data master model
 - 9.2 Pada bagian master model harus memiliki fungsi untuk merubah data master model
 - 9.3 Pada bagian master model harus memiliki fungsi untuk melihat daftar master model
10. Daftar Harga Beli
 - 10.1 Pada bagian daftar harga beli harus memiliki fungsi untuk menambah data daftar harga beli
 - 10.2 Pada bagian daftar harga beli harus memiliki fungsi untuk merubah data harga beli
 - 10.3 Pada bagian daftar harga beli harus memiliki fungsi untuk melihat daftar harga beli
11. Daftar Harga Jual
 - 11.1 Pada bagian daftar harga jual harus memiliki fungsi untuk menambah data daftar harga jual
 - 11.2 Pada bagian daftar harga jual harus memiliki fungsi untuk merubah data harga jual

11.3 Pada bagian daftar harga jual harus memiliki fungsi untuk melihat daftar harga jual

4.3 Identifikasi Aktor

Pada tahap ini akan ditunjukkan aktor yang berinteraksi dengan sistem. Aktor digunakan untuk membantu memberi gambaran mengenai apa yang dapat dilakukan aktor dalam menggunakan sistem. Keterangan mengenai aktor pengguna dapat dilihat pada tabel Tabel 4.1

Tabel 4.1 Identifikasi Aktor

Nama Aktor	Deskripsi Aktor
User	User merupakan aktor yang dapat melakukan aktifitas login untuk masuk ke dalam sistem
Admin	Admin merupakan aktor yang digunakan oleh Sales Counter maupun Manager untuk melakukan manajemen master data. Admin dapat melakukan manajemen master data, baik dari melakukan penambahan data, pembaharuan data hingga penghapusan data apabila diperlukan. Master data yang dimaksud adalah data data yang berisi informasi detail tentang model sepeda motor dan juga data yang menentukan harga jual pada dealer PT. Mega Abadi Motorindo
Sales Counter	Sales counter merupakan aktor yang dapat melakukan permintaan barang kepada gudang pusat yang ada di Jakarta. Sales counter juga berkewajiban memeriksa barang yang telah datang apakah sesuai dengan barang yang telah diminta. Sales counter juga dapat melayani penjualan barang terhadap pelanggan, mulai dari proses penjualan hingga penerbitan invoice. Sales dapat mengirimkan informasi baik ke masing masing pelanggan maupun semua pelanggan terkait info, promo dan kebutuhan lain yang mendukung pelayanan terhadap pelanggan.
Kasir	Kasir merupakan aktor yang dapat melakukan pengelolaan kas masuk dan kas keluar pada PT. Mega Abadi Motorindo. Kasir bertanggung jawab atas keuangan yang ada pada PT. Mega Abadi Motorindo. Untuk mendukung dan memudahkan tanggung jawab tersebut, kasir juga dapat melihat history transaksi yang pernah terjadi pada PT. Mega Abadi Motorindo
Mekanik	Mekanik merupakan aktor yang dapat melakukan registrasi antrian servis motor pada pelanggan yang akan melakukan servis. Sebelum motor masuk ke ruangan servis, mekanik akan mendata motor yang akan di servis. Mekanik disini juga dapat melihat riwayat pada saat melakukan registrasi, apakah motor yang dimiliki pelanggan masih memiliki layanan gratis servis.
Manager	Manager merupakan aktor yang dapat melihat seluruh history transaksi yang pernah terjadi pada PT. Mega Abadi Motorindo. Manager juga dapat melihat berapa jumlah stok yang ada pada gudang dealer PT. Mega Abadi Motorindo

4.4 Spesifikasi Kebutuhan

Software Requirement Specification (SRS) digunakan dalam proses identifikasi kebutuhan. Fungsi dari SRS adalah untuk menampung daftar kebutuhan sistem agar mudah dimengerti. Kebutuhan yang ditampung dengan SRS adalah kebutuhan fungsional dan kebutuhan non fungsional. Tahap pertama dalam identifikasi kebutuhan adalah untuk mendapatkan daftar kebutuhan awal yang digunakan dalam aplikasi. Setelah dibangun prototype berdasarkan daftar kebutuhan, kemudian pada tahap kedua dilakukan evaluasi pengguna untuk mendapatkan data kebutuhan atau perubahan data sesuai hasil evaluasi pengguna agar aplikasi sesuai dengan kebutuhan pengguna.

4.4.1 Daftar Kebutuhan Fungsional

Kebutuhan fungsional merupakan suatu kebutuhan yang harus disediakan oleh sistem. Spesifikasi kebutuhan fungsional ditunjukkan pada Tabel 4.2. Setiap kebutuhan akan diberikan kode SAT_F_XXYY. SAT merupakan singkatan dari Sistem Aplikasi Terintegrasi, F merupakan kebutuhan fungsional dan XX merupakan nomer dari definisi kebutuhan utama dan YY menunjukkan nomer spesifikasi dari definisi kebutuhan utama.

Tabel 4.2 Daftar Kebutuhan Fungsional dan Pemetaan Nama Use Case

No	Kebutuhan Sistem	Kode	Use Case	Actor
1	Sistem harus menyediakan fungsi login untuk mengizinkan pengguna mengakses sistem sesuai dengan hak aksesnya dengan cara memasukkan username dan password	SAT_F_0100	Login	User
2	Sistem harus menyediakan menu logout agar pengguna yang sudah masuk ke dalam sistem dapat keluar dari sistem	SAT_F_0200	Logout	Admin, Sales Counter, Kasir, Mekanik, Manager
3	Sistem dapat menampilkan daftar master data motor yang telah ada pada database	SAT_F_0300	Melihat master data motor	Admin
3.1	Sistem menyediakan informasi seperti model code, model name, engine code, model basic code, transmission type, cylinder type, status, model colour code, model year	SAT_F_0301		
4	Sistem harus dapat membuat master data motor baru yang	SAT_F_0400	Membuat master data motor	Admin

	hanya bisa diakses oleh Admin			
4.1	Sistem menyediakan form berupa model code,model name, engine code, model basic code, transmission type, cylinder type, status, model colour code, model year	SAT_F_0401		
5	Sistem harus dapat mengubah master data motor yang hanya bisa diakses oleh Admin	SAT_F_0500	Mengubah master data motor	Admin
6	Sistem harus dapat menghapus master data motor yang hanya bisa diakses oleh Admin	SAT_F_0600	Menghapus master data motor	Admin
6.1	Sistem menyediakan tombol untuk melakukan penghapusan data master motor	SAT_F_0601		
7	Sistem dapat menampilkan daftar harga beli	SAT_F_0700	Melihat daftar harga beli	Admin
7.1	Sistem menyediakan informasi seperti model code, model year dan DPP	SAT_F_0701		
8	Sistem harus dapat menambah daftar harga beli yang hanya bisa diakses oleh Admin	SAT_F_0800	Menambah daftar harga beli	Admin
8.1	Sistem menyediakan form berupa model code, model year dan DPP	SAT_F_0801		
9	Sistem harus dapat mengubah daftar harga beli yang hanya bisa diakses oleh Admin	SAT_F_0900	Mengubah daftar harga beli	Admin
10	Sistem harus dapat menghapus daftar harga beli yang hanya bisa diakses oleh Admin	SAT_F_1000	Menghapus daftar harga beli	Admin
10.1	Sistem menyediakan tombol untuk melakukan penghapusan data harga beli	SAT_F_1001		
11	Sistem dapat menampilkan daftar harga jual	SAT_F_1100	Melihat daftar harga jual	Admin
11.1	Sistem menyediakan informasi seperti group price, model code, model year.	SAT_F_1101		

12	Sistem harus dapat menambah daftar harga jual yang hanya bisa diakses oleh Admin	SAT_F_1200	Menambah daftar harga jual	Admin
12.1	Sistem menyediakan form berupa group price, model code, model year.	SAT_F_1201		
13	Sistem harus dapat mengubah daftar harga jual yang hanya bisa diakses oleh Admin	SAT_F_1300	Mengubah daftar harga jual	Admin
14	Sistem harus dapat menghapus daftar harga jual yang hanya bisa diakses oleh Admin	SAT_F_1400	Menghapus daftar harga jual	Admin
14.1	Sistem menyediakan tombol untuk melakukan penghapusan data harga jual.	SAT_F_1401		
15	Sistem dapat menampilkan permintaan barang telah dilakukan	SAT_F_1500	Melihat daftar permintaan barang	Sales Counter
16	Sistem harus dapat membuat permintaan barang baru yang hanya bisa diakses oleh Sales Counter	SAT_F_1600	Membuat permintaan barang baru	Sales Counter
17	Sistem harus dapat mengubah permintaan barang baru yang hanya bisa diakses oleh Sales Counter	SAT_F_1700	Mengubah permintaan barang baru	Sales Counter
18	Sistem harus dapat menghapus permintaan barang yang hanya bisa diakses oleh Sales Counter	SAT_F_1800	Menghapus permintaan barang baru	Sales Counter
19	Sistem harus dapat mengirimkan informasi permintaan barang baru ke kantor pusat	SAT_F_1900	Mengirim informasi permintaan barang baru	Sales Counter
19.1	Sistem menyediakan tombol untuk melakukan pengiriman permintaan barang baru	SAT_F_1901		
19.2	Sistem akan mengirimkan melalui email menuju kantor pusat	SAT_F_1902		
20	Sistem harus dapat membuat bukti penerimaan barang yang hanya bisa diakses oleh Sales Counter	SAT_F_2000	Membuat bukti penerimaan barang	Sales Counter

20.1	Sistem menyediakan form berupa no BPU, no PO, no ref dan gudang	SAT_F_2001		
21	Sistem dapat menampilkan riwayat bukti penerimaan barang	SAT_F_2100	Melihat bukti penerimaan barang	Sales Counter
22	Sistem dapat mengubah bukti penerimaan barang yang hanya dapat diakses oleh Sales Counter	SAT_F_2200	Mengubah bukti penerimaan barang	Sales Counter
23	Sistem dapat menghapus bukti penerimaan barang yang hanya dapat diakses oleh Sales Counter	SAT_F_2300	Menghapus bukti penerimaan barang	Sales Counter
24	Sistem dapat mengunci status bukti penerimaan barang yang hanya dapat diakses oleh Sales Counter	SAT_F_2400	Mengunci status bukti penerimaan barang	Sales Counter
24.1	Sistem menyediakan tombol untuk mengunci bukti penerimaan	SAT_F_2401		
25	Sistem dapat menampilkan jumlah stok barang yang tersedia	SAT_F_2500	Melihat jumlah stok barang	Sales Counter
25.1	Sistem menyediakan informasi seperti tanggal, model code, model year, model color, no rangka, no mesin dan refrensi no bpk	SAT_F_2501		
26	Sistem harus dapat membuat order penjualan baru yang hanya bisa diakses oleh Sales Counter	SAT_F_2600	Membuat order penjualan baru	Sales Counter
26.1	Sistem menyediakan form berupa no SO, pelanggan, salesman, tagihan, gudang, group price	SAT_F_2601		
27	Sistem dapat menampilkan order penjualan yang hanya bisa diakses oleh Sales Counter	SAT_F_2700	Melihat order penjualan	Sales Counter
28	Sistem dapat mengubah order penjualan yang hanya bisa diakses oleh Sales Counter	SAT_F_2800	Mengubah order penjualan	Sales Counter
29	Sistem dapat menghapus order penjualan yang hanya bisa diakses oleh Sales Counter	SAT_F_2900	Menghapus order penjualan	Sales Counter

30	Sistem dapat detail order penjualan yang hanya bisa diakses oleh Sales Counter	SAT_F_3000	Mencetak detail order penjualan	Sales Counter
30.1	Sistem menyediakan tombol untuk mencetak detail order penjualan	SAT_F_3001		
30.2	Sistem menyediakan informasi seperti nama sales, tipe sales, pelanggan, no so, tanggal order, dan detail motor	SAT_F_3002		
31	Sistem harus dapat membuat surat jalan yang hanya bisa diakses oleh Sales Counter	SAT_F_3100	Membuat surat jalan	Sales Counter
32	Sistem dapat mengubah surat jalan yang hanya bisa diakses oleh Sales Counter	SAT_F_3200	Mengubah surat jalan	Sales Counter
33	Sistem dapat melihat surat jalan yang hanya bisa diakses oleh Sales Counter	SAT_F_3300	Melihat surat jalan	Sales Counter
34	Sistem dapat menghapus surat jalan yang hanya bisa diakses oleh Sales Counter	SAT_F_3400	Menghapus surat jalan	Sales Counter
35	Sistem dapat mencetak surat jalan yang hanya bisa diakses oleh Sales Counter	SAT_F_3500	Mencetak surat jalan	Sales Counter
35.1	Sistem menyediakan tombol untuk mencetak surat jalan	SAT_F_3501		
35.2	Sistem menyediakan informasi seperti detail motor	SAT_F_3502		
36	Sistem harus dapat membuat surat faktur pajak yang hanya bisa diakses oleh Sales Counter	SAT_F_3600	Membuat surat faktur pajak	Sales Counter
37	Sistem dapat mengubah surat faktur pajak yang hanya bisa diakses oleh Sales Counter	SAT_F_3700	Mengubah surat faktur pajak	Sales Counter
38	Sistem dapat melihat surat faktur pajak yang hanya bisa diakses oleh Sales Counter	SAT_F_3800	Melihat surat faktur pajak	Sales Counter
39	Sistem dapat menghapus surat faktur pajak yang hanya bisa diakses oleh Sales Counter	SAT_F_3900	Menghapus surat faktur pajak	Sales Counter
40	Sistem dapat mencetak surat faktur pajak yang hanya bisa diakses oleh Sales Counter	SAT_F_4000	Mencetak surat faktur pajak	Sales Counter

40.1	Sistem menyediakan tombol untuk mencetak surat faktur pajak	SAT_F_4001		
40.2	Sistem menyediakan informasi seperti detail motor	SAT_F_4002		
41	Sistem harus dapat membuat invoice yang hanya bisa diakses oleh Sales Counter	SAT_F_4100	Membuat invoice	Sales Counter
41.1	Sistem menyediakan form berupa tagihan kepada dan keterangan	SAT_F_4101		
42	Sistem dapat membuat invoice yang hanya bisa diakses oleh Sales Counter	SAT_F_4200	Mengubah invoice	Sales Counter
43	Sistem dapat melihat invoice yang hanya bisa diakses oleh Sales Counter	SAT_F_4300	Melihat invoice	Sales Counter
44	Sistem dapat menghapus invoice yang hanya bisa diakses oleh Sales Counter	SAT_F_4400	Menghapus invoice	Sales Counter
45	Sistem dapat menambah sales order pada invoice yang hanya bisa diakses oleh Sales Counter	SAT_F_4500	Menambah sales order pada invoice	Sales Counter
46	Sistem dapat mencetak invoice yang hanya bisa diakses oleh Sales Counter	SAT_F_4600	Mencetak detail invoice	Sales Counter
46.1	Sistem menyediakan tombol untuk mencetak detail order penjualan	SAT_F_4601		
46.2	Sistem menyediakan informasi seperti tagihan dan detail sales order	SAT_F_4602		
47	Sistem dapat membuat master data pelanggan yang hanya bisa diakses oleh Sales Counter	SAT_F_4700	Membuat master data pelanggan	Sales Counter
47.1	Sistem menyediakan form berupa kode pelanggan, nama pelanggan, insial pelanggan, gender, no hp, no telepon, alamat email, tanggal lahir, alamat, kode pos, kota, provinsi, status	SAT_F_4701		
48	Sistem dapat mengubah master data pelanggan yang	SAT_F_4800	Mengubah master data pelanggan	Sales Counter

	hanya bisa diakses oleh Sales Counter			
49	Sistem dapat melihat master data pelanggan yang hanya bisa diakses oleh Sales Counter	SAT_F_4900	Melihat master data pelanggan	Sales Counter
50	Sistem dapat menghapus master data pelanggan yang hanya bisa diakses oleh Sales Counter	SAT_F_5000	Menghapus master data pelanggan	Sales Counter
51	Sistem dapat membuat master data leasing yang hanya bisa diakses oleh Sales Counter	SAT_F_5100	Membuat master data leasing	Sales Counter
51.1	Sistem menyediakan form berupa kode leasing dan deskripsi leasing	SAT_F_5101		
52	Sistem dapat mengubah master data leasing yang hanya bisa diakses oleh Sales Counter	SAT_F_5200	Mengubah master data leasing	Sales Counter
53	Sistem dapat melihat master data leasing yang hanya bisa diakses oleh Sales Counter	SAT_F_5300	Melihat master data leasing	Sales Counter
54	Sistem dapat menghapus master data leasing yang hanya bisa diakses oleh Sales Counter	SAT_F_5400	Menghapus master data leasing	Sales Counter
55	Sistem dapat membuat data kas master yang hanya bisa diakses oleh Kasir	SAT_F_5500	Membuat data kas master	Kasir
55.1	Sistem menyediakan form berupa idkas, nama kas, jumlah	SAT_F_5501		
56	Sistem dapat mengubah data kas master yang hanya bisa diakses oleh Kasir	SAT_F_5600	Mengubah data kas master	Kasir
57	Sistem dapat melihat data kas master yang hanya bisa diakses oleh Kasir	SAT_F_5700	Melihat data kas master	Kasir
58	Sistem dapat menghapus data kas master yang hanya bisa diakses oleh Kasir	SAT_F_5800	Menghapus data kas master	Kasir
59	Sistem dapat membuat data kas terima yang hanya bisa diakses oleh Kasir	SAT_F_5900	Membuat data kas terima	Kasir

59.1	Sistem menyediakan form berupa penerimaan, no bkt, tanggal, kas, no invoice, atas nama, jumlah terima, keterangan	SAT_F_5901		
60	Sistem dapat mengubah data kas terima yang hanya bisa diakses oleh Kasir	SAT_F_6000	Mengubah data kas terima	Kasir
61	Sistem dapat melihat data kas terima yang hanya bisa diakses oleh Kasir	SAT_F_6100	Melihat data kas terima	Kasir
62	Sistem dapat menghapus data kas terima yang hanya bisa diakses oleh Kasir	SAT_F_6200	Menghapus data kas terima	Kasir
63	Sistem dapat membuat data kas keluar yang hanya bisa diakses oleh Kasir	SAT_F_6300	Membuat data kas keluar	Kasir
63.1	Sistem menyediakan form berupa pengeluaran, no bkk, tanggal, kas, atas nama, jumlah keluar, keperluan, keterangan	SAT_F_6301		
64	Sistem dapat mengubah data kas keluar yang hanya bisa diakses oleh Kasir	SAT_F_6400	Mengubah data kas keluar	Kasir
65	Sistem dapat melihat data kas keluar yang hanya bisa diakses oleh Kasir	SAT_F_6500	Melihat data kas keluar	Kasir
66	Sistem dapat menghapus data kas keluar yang hanya bisa diakses oleh Kasir	SAT_F_6600	Menghapus data kas keluar	Kasir
67	Sistem dapat melihat riwayat kas terima	SAT_F_6700	Melihat riwayat kas terima	Manager
68	Sistem dapat melihat riwayat kas keluar	SAT_F_6800	Melihat riwayat kas keluar	Manager
69	Sistem dapat melihat riwayat sales order	SAT_F_6900	Melihat riwayat sales order	Manager
70	Sistem dapat membuat registrasi servis yang hanya diakses oleh Mekanik	SAT_F_7000	Menambah data antrian servis	Mekanik
70.1	Sistem menyediakan form berupa no ref, pelanggan, no rangka, no mesin, no polisi, status	SAT_F_7001		
71	Sistem dapat mengubah registrasi servis yang hanya diakses oleh Mekanik	SAT_F_7100	Mengubah data antrian servis	Mekanik

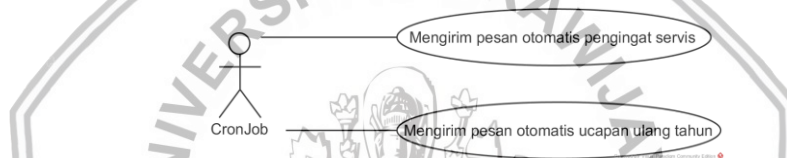
72	Sistem dapat melihat registrasi servis yang hanya diakses oleh Mekanik	SAT_F_7200	Melihat data antrian servis	Mekanik
73	Sistem dapat menghapus registrasi servis yang hanya diakses oleh Mekanik	SAT_F_7300	Menghapus data antrian servis	Mekanik
74	Sistem dapat merubah status antrian servis yang hanya diakses oleh Mekanik	SAT_F_7400	Merubah status antrian servis	Mekanik
75	Sistem dapat melakukan pengecekan gratis servis yang hanya diakses oleh Mekanik	SAT_F_7500	Melakukan pengecekan gratis servis	Mekanik
75.1	Sistem menyediakan form berupa no rangka dan no mesin	SAT_F_7501		
76	Sistem dapat melihat riwayat pesan kepada pelanggan yang hanya bisa diakses oleh Sales Counter	SAT_F_7600	Melihat riwayat pesan kepada pelanggan	Sales Counter
77	Sistem dapat mengirim pesan kepada salah satu pelanggan yang hanya bisa diakses oleh Sales Counter	SAT_F_7700	Mengirim pesan kepada pelanggan	Sales Counter
77.1	Sistem menyediakan form berupa nama pelanggan, no tujuan, pesan dan note	SAT_F_7701		
78	Sistem dapat mengirim pesan kepada semua pelanggan yang hanya bisa diakses oleh Sales Counter	SAT_F_7800	Mengirim pesan kepada banyak pelanggan	Sales Counter
78.1	Sistem menyediakan form berupa pesan dan note	SAT_F_7801		
79	Sistem dapat mengirim pesan secara otomatis kepada pelanggan untuk mengingatkan waktu servis berkala	SAT_F_7900	Mengirim pesan otomatis pengingat servis	CronJob
80	Sistem dapat mengirim pesan secara otomatis kepada pelanggan ketika pelanggan sedang berulang tahun	SAT_F_8000	Mengirim pesan otomatis ucapan ulang tahun	CronJob
81	Sistem dapat mengunci status invoice yang hanya dapat diakses oleh Sales Counter	SAT_F_8100	Mengunci status invoice	Sales Counter

4.5 Pemodelan Kebutuhan

Untuk mempermudah memahami kebutuhan sistem, maka dibuat suatu pemodelan kebutuhan. Dalam pemodelan kebutuhan, kebutuhan fungsional yang telah terdefinisi akan diubah menjadi *use case diagram* dan *Use Case Scenario*. *Use case diagram* dibuat dengan tujuan untuk menjelaskan apa saja yang dapat dilakukan oleh aktor dalam sistem ini. Setelah *use case diagram* akan dibuat *Use Case Scenario* untuk memperjelas rincian tiap case dari *use case diagram* sebelumnya.

4.5.1 Use Case Diagram

Use case diagram merupakan diagram yang berfungsi menunjukkan hal-hal apa saja yang dapat dilakukan oleh aktor berdasarkan kebutuhan fungsional yang telah di deskripsikan. Berdasarkan kebutuhan fungsional yang telah di deskripsikan maka dibuat use case diagram pada gambar 4.1 dan gambar 4.2



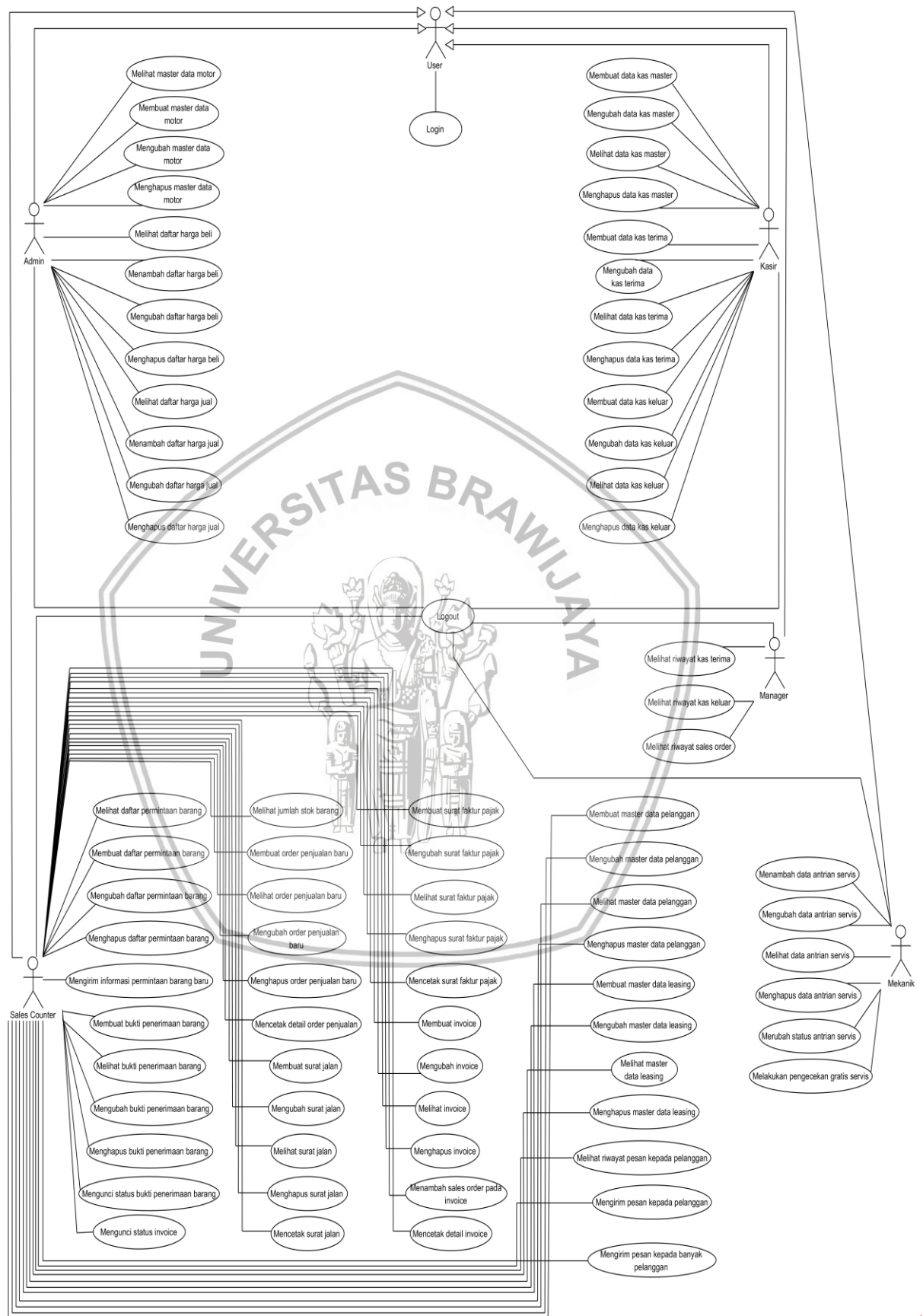
Gambar 4.1 Use Case Diagram Sistem Eksternal

4.5.2 Use Case Scenario

Use Case Scenario merupakan urutan alur dari use case yang sudah didefinisikan. Usecase scenario terdiri dari *Actor*, *Objective*, *Pre-Condition*, *Main Flow*, *Alternative Flow*, dan *Post-Condition*. Setiap *Use Case Scenario* menggambarkan setiap use case yang sudah dibuat. Sehingga diperoleh 81 *Use Case Scenario*.

Tabel 4.3 Use Case Scenario Login

Flow of Event untuk Login	
Kode Kebutuhan	SAT_F_0100
Objective	Use case ini berguna untuk masuk kedalam sistem
Actor	User
Pre-condition	Aktor masuk ke halaman login
Main Flow	<ol style="list-style-type: none"> 1. Aktor memasukkan username 2. Aktor memasukkan password 3. Aktor menekan tombol sign in 4. Sistem melakukan pengecekan data username dan password pada database
Alternatif Flow	<ol style="list-style-type: none"> 1. Jika username atau password salah, maka akan kembali ke halaman login
Post-condition	Aktor berhasil masuk ke dalam sistem sesuai dengan statusnya



Gambar 4.2 Use Case Diagram Sistem Utama

Tabel 4.4 Use Case Scenario Logout

<i>Flow of Event untuk Logout</i>	
Kode Kebutuhan	SAT_F_0200
Objective	Use case ini berguna untuk keluar dari sistem
Actor	Admin, Sales Counter, Kasir, Mekanik, Manager
Pre-condition	Aktor sudah berada didalam sistem
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu logout 2. Sistem melakukan destroy session user
Alternatif Flow	-
Post-condition	Aktor berhasil keluar dari sistem

Tabel 4.5 Use Case Scenario melihat master data motor

<i>Flow of Event untuk melihat master data motor</i>	
Kode Kebutuhan	SAT_F_0300
Objective	Use case ini berguna untuk melihat master data motor
Actor	Admin
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu master data motor 2. Sistem menampilkan master data motor dalam bentuk tabel
Alternatif Flow	Jika data dari master data motor tidak ada, maka akan ditampilkan "tidak ada data"
Post-condition	Data master motor ditampilkan pada tabel

Tabel 4.6 Use Case Scenario membuat master data motor

<i>Flow of Event untuk membuat master data motor</i>	
Kode Kebutuhan	SAT_F_0400
Objective	Use case ini berguna untuk membuat master data motor
Actor	Admin
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu master data motor.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ master data" 2. Sistem akan menampilkan modal 3. Aktor mengisi input data yang diperlukan 4. Sistem akan menampilkan popup "Input master data baru sukses"
Alternatif Flow	Jika data sudah ada maka akan menampilkan popup "master data sudah ada"
Post-condition	Data master motor telah tersimpan oleh sistem

Tabel 4.7 Use Case Scenario mengubah master data motor

<i>Flow of Event untuk mengubah master data motor</i>	
Kode Kebutuhan	SAT_F_0500
Objective	Use case ini berguna untuk mengubah master data motor
Actor	Admin
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu master data motor.

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik ikon pensil pada tabel 2. Sistem akan menampilkan modal 3. Aktor memperbaharui data pada form, dan mengklik tombol update 4. Sistem akan menampilkan popup "Edit master data motor sukses"
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data master motor telah diperbaharui oleh sistem

Tabel 4.8 Use Case Scenario menghapus master data motor

<i>Flow of Event untuk</i> menghapus master data motor	
Kode Kebutuhan	SAT_F_0600
<i>Objective</i>	Use case ini berguna untuk menghapus master data motor
<i>Actor</i>	Admin
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu master data motor.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik ikon sampah pada tabel 2. Sistem akan menampilkan modal peringatan 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan popup "Master data motor telah terhapus"
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus master data motor yang dipilih dari database

Tabel 4.9 Use Case Scenario melihat daftar harga beli

<i>Flow of Event untuk</i> melihat daftar harga beli	
Kode Kebutuhan	SAT_F_0700
<i>Objective</i>	Use case ini berguna untuk melihat daftar harga beli
<i>Actor</i>	Admin
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu daftar harga beli 2. Sistem menampilkan data daftar harga beli pada tabel
<i>Alternatif Flow</i>	Jika data dari daftar harga beli tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data harga beli ditampilkan pada tabel

Tabel 4.10 Use Case Scenario menambah daftar harga beli

<i>Flow of Event untuk</i> menambah daftar harga beli	
Kode Kebutuhan	SAT_F_0800
<i>Objective</i>	Use case ini berguna untuk menambah daftar harga beli
<i>Actor</i>	Admin
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu daftar harga beli.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ daftar harga beli" 2. Sistem akan menampilkan modal 3. Aktor akan mengisi form dengan data data yang dibutuhkan, lalu mengklik tombol "submit" 4. Sistem akan menampilkan popup "Daftar harga beli baru telah ditambahkan"

<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data harga beli telah tersimpan oleh sistem

Tabel 4.11 Use Case Scenario mengubah daftar harga beli

<i>Flow of Event untuk</i> mengubah daftar harga beli	
Kode Kebutuhan	SAT_F_0900
<i>Objective</i>	Use case ini berguna untuk mengubah daftar harga beli
<i>Actor</i>	Admin
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu daftar harga beli.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon pensil pada tabel 2. Sistem akan menampilkan modal 3. Aktor akan mengisi form dengan data data yang akan dirubah, lalu mengklik tombol "submit" 4. Sistem akan menampilkan popup "Daftar harga beli baru telah dirubah"
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data harga beli telah diperbaharui oleh sistem

Tabel 4.12 Use Case Scenario menghapus daftar harga beli

<i>Flow of Event untuk</i> menghapus daftar harga beli	
Kode Kebutuhan	SAT_F_1000
<i>Objective</i>	Use case ini berguna untuk menghapus daftar harga beli
<i>Actor</i>	Admin
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu daftar harga beli.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan popup "Daftar harga beli baru telah dihapus"
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus daftar harga beli yang dipilih dari database

Tabel 4.13 Use Case Scenario melihat daftar harga jual

<i>Flow of Event untuk</i> melihat daftar harga jual	
Kode Kebutuhan	SAT_F_1100
<i>Objective</i>	Use case ini berguna untuk melihat daftar harga jual
<i>Actor</i>	Admin
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor telah memilih menu daftar harga jual 2. Sistem menampilkan data daftar harga jual pada tabel
<i>Alternatif Flow</i>	Jika data dari daftar harga jual tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data harga jual ditampilkan pada tabel

Tabel 4.14 Use Case Scenario menambah daftar harga jual

<i>Flow of Event untuk</i> menambah daftar harga jual	
-------------------------------------------------------	--

Kode Kebutuhan	SAT_F_1200
Objective	Use case ini berguna untuk menambah daftar harga jual
Actor	Admin
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu daftar harga jual.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ daftar harga jual" 2. Sistem akan menampilkan modal 3. Aktor akan mengisi form dengan data data yang dibutuhkan, lalu mengklik tombol "submit" 4. Sistem akan menampilkan popup "Daftar harga jual baru telah ditambahkan"
Alternatif Flow	-
Post-condition	Data harga jual telah tersimpan oleh sistem

Tabel 4.15 Use Case Scenario mengubah daftar harga jual

Flow of Event untuk mengubah daftar harga jual	
Kode Kebutuhan	SAT_F_1300
Objective	Use case ini berguna untuk mengubah daftar harga jual
Actor	Admin
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu daftar harga jual.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon pensil pada tabel 2. Sistem akan menampilkan modal 3. Aktor akan mengisi form dengan data data yang akan dirubah, lalu mengklik tombol "submit" 4. Sistem akan menampilkan popup "Daftar harga jual baru telah dirubah"
Alternatif Flow	-
Post-condition	Data harga jual telah diperbaharui oleh sistem

Tabel 4.16 Use Case Scenario menghapus daftar harga jual

Flow of Event untuk menghapus daftar harga jual	
Kode Kebutuhan	SAT_F_1400
Objective	Use case ini berguna untuk menghapus daftar harga jual
Actor	Admin
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai admin dan telah memilih menu daftar harga jual.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan popup "Daftar harga jual baru telah dihapus"
Alternatif Flow	-
Post-condition	Sistem akan menghapus daftar harga jual yang dipilih dari database

Tabel 4.17 Use Case Scenario melihat daftar permintaan barang

Flow of Event untuk melihat daftar permintaan barang	
Kode Kebutuhan	SAT_F_1500
Objective	Use case ini berguna untuk melihat daftar permintaan barang

<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu daftar permintaan barang. 2. Sistem menampilkan daftar melihat daftar permintaan barang dalam tabel
<i>Alternatif Flow</i>	Jika data pada daftar permintaan barang tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data permintaan barang ditampilkan pada tabel

Tabel 4.18 Use Case Scenario membuat permintaan barang baru

<i>Flow of Event untuk membuat permintaan barang baru</i>	
Kode Kebutuhan	SAT_F_1600
<i>Objective</i>	Use case ini berguna untuk membuat permintaan barang baru
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar permintaan barang.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ Permintaan Baru" 2. Sistem akan menampilkan modal 3. Aktor akan mengisi input dari data yang diperlukan, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan popup "Permintaan barang baru ke gudang sukses"
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data permintaan barang baru telah tersimpan oleh sistem

Tabel 4.19 Use Case Scenario mengubah permintaan barang baru

<i>Flow of Event untuk mengubah permintaan barang baru</i>	
Kode Kebutuhan	SAT_F_1700
<i>Objective</i>	Use case ini berguna untuk mengubah permintaan barang baru
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar permintaan barang.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit barang baru. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar permintaan barang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data permintaan barang telah diperbaharui oleh sistem

Tabel 4.20 Use Case Diagram menghapus permintaan barang baru

<i>Flow of Event untuk menghapus permintaan barang baru</i>	
Kode Kebutuhan	SAT_F_1800
<i>Objective</i>	Use case ini berguna untuk menghapus permintaan barang baru
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar permintaan barang.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel

	<ol style="list-style-type: none"> 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan daftar permintaan barang yang baru
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus daftar permintaan barang yang dipilih dari database

Tabel 4.21 Use Case Scenario mengirim informasi permintaan barang

<i>Flow of Event untuk mengirim informasi permintaan barang</i>	
Kode Kebutuhan	SAT_F_1900
<i>Objective</i>	Use case ini berguna untuk mengirim informasi permintaan barang
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar permintaan barang.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Send Email" pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya"
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data permintaan barang terkirim melalui email

Tabel 4.22 Use Case Scenario membuat bukti penerimaan barang

<i>Flow of Event untuk membuat bukti penerimaan barang</i>	
Kode Kebutuhan	SAT_F_2000
<i>Objective</i>	Use case ini berguna untuk membuat bukti penerimaan barang
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar penerimaan barang.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik ikon "chevron" pada tabel 2. Sistem akan menampilkan modal 3. Aktor akan mengisi input form yang dibutuhkan, lalu mengklik tombol "submit" 4. Sistem akan menampilkan daftar bukti penerimaan barang
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data penerimaan barang akan tersimpan oleh sistem

Tabel 4.23 Use Case Scenario melihat bukti penerimaan barang

<i>Flow of Event untuk melihat bukti penerimaan barang</i>	
Kode Kebutuhan	SAT_F_2100
<i>Objective</i>	Use case ini berguna untuk melihat bukti penerimaan barang
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu daftar penerimaan barang. 2. Sistem menampilkan data bukti penerimaan barang pada tabel
<i>Alternatif Flow</i>	Jika data bukti penerimaan barang tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data penerimaan barang ditampilkan pada tabel

Tabel 4.24 Use Case Scenario mengubah bukti penerimaan barang

<i>Flow of Event</i> untuk mengubah bukti penerimaan barang	
Kode Kebutuhan	SAT_F_2200
Objective	Use case ini berguna untuk mengubah bukti penerimaan barang
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar penerimaan barang.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar bukti penerimaan barang
Alternatif Flow	-
Post-condition	Data penerimaan barang telah diperbaharui oleh sistem

Tabel 4.25 Use Case Scenario menghapus bukti penerimaan barang

<i>Flow of Event</i> untuk menghapus bukti penerimaan barang	
Kode Kebutuhan	SAT_F_2300
Objective	Use case ini berguna untuk menghapus bukti penerimaan barang
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar penerimaan barang.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan daftar bukti penerimaan barang yang baru
Alternatif Flow	-
Post-condition	Sistem akan menghapus bukti penerimaan barang yang dipilih dari database

Tabel 4.26 Use Case Scenario mengunci status bukti penerimaan barang

<i>Flow of Event</i> untuk mengunci status bukti penerimaan barang	
Kode Kebutuhan	SAT_F_2400
Objective	Use case ini berguna untuk mengunci status bukti penerimaan barang
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu daftar penerimaan barang.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Lock" pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya"
Alternatif Flow	-
Post-condition	Status bukti penerimaan barang akan menjadi terkunci dan tidak dapat dirubah

Tabel 4.27 Use Case Scenario melihat jumlah stok barang

<i>Flow of Event</i> untuk melihat jumlah stok barang	
Kode Kebutuhan	SAT_F_2500

<i>Objective</i>	Use case ini berguna untuk melihat jumlah stok barang
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu stok barang 2. Sistem menampilkan jumlah stok barang
<i>Alternatif Flow</i>	Jika data pada stok barang tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data stok barang ditampilkan pada tabel

Tabel 4.28 Use Case Scenario membuat order penjualan baru

<i>Flow of Event untuk membuat order penjualan baru</i>	
Kode Kebutuhan	SAT_F_2600
<i>Objective</i>	Use case ini berguna untuk membuat order penjualan baru
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu sales order.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ order baru" 2. Sistem menampilkan modal 3. Aktor mengisi input dengan data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan daftar penjualan order baru
<i>Alternatif Flow</i>	
<i>Post-condition</i>	Data order penjualan telah tersimpan oleh sistem

Tabel 4.29 Use Case Scenario melihat order penjualan

<i>Flow of Event untuk melihat order penjualan</i>	
Kode Kebutuhan	SAT_F_2700
<i>Objective</i>	Use case ini berguna untuk melihat order penjualan
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu sales order 2. Sistem menampilkan order penjualan pada tabel
<i>Alternatif Flow</i>	Jika data pada order penjualan tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data order penjualan ditampilkan pada tabel

Tabel 4.30 Use Case Scenario mengubah order penjualan

<i>Flow of Event untuk mengubah order penjualan</i>	
Kode Kebutuhan	SAT_F_2800
<i>Objective</i>	Use case ini berguna untuk mengubah order penjualan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu sales order.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit"

	4. Sistem akan menampilkan daftar order penjualan
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data order penjualan telah diperbaharui oleh sistem

Tabel 4.31 Use Case Scenario menghapus order penjualan

<i>Flow of Event untuk menghapus order penjualan</i>	
Kode Kebutuhan	SAT_F_2900
<i>Objective</i>	Use case ini berguna untuk menghapus order penjualan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu sales order.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan daftar order penjualan yang baru
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus order penjualan yang dipilih dari database

Tabel 4.32 Use Case Scenario mencetak order penjualan

<i>Flow of Event untuk mencetak order penjualan</i>	
Kode Kebutuhan	SAT_F_3000
<i>Objective</i>	Use case ini berguna untuk mencetak order penjualan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu sales order.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon print pada tabel 2. Sistem akan menampilkan halaman popup 3. Sistem akan menampilkan daftar order penjualan yang akan dicetak dalam bentuk pdf
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Detail daftar order penjualan dalam bentuk pdf

Tabel 4.33 Use Case Scenario membuat surat jalan

<i>Flow of Event untuk membuat surat jalan</i>	
Kode Kebutuhan	SAT_F_3100
<i>Objective</i>	Use case ini berguna untuk membuat surat jalan
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu delivery order.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ surat jalan" 2. Sistem menampilkan halaman input data 3. Aktor memasukkan nomer "Order Penjualan" beserta data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan hasil
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data surat jalan telah tersimpan oleh sistem

Tabel 4.34 Use Case Scenario melihat surat jalan

Flow of Event untuk melihat surat jalan	
Kode Kebutuhan	SAT_F_3300
Objective	Use case ini berguna untuk melihat surat jalan
Actor	Sales counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu delivery order. 2. Sistem menampilkan daftar surat jalan pada tabel
Alternatif Flow	Jika data pada surat jalan tidak ada, maka akan ditampilkan "tidak ada data"
Post-condition	Data surat jalan ditampilkan pada tabel

Tabel 4.35 Use Case Scenario mengubah surat jalan

Flow of Event untuk mengubah surat jalan	
Kode Kebutuhan	SAT_F_3200
Objective	Use case ini berguna untuk mengubah surat jalan
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu delivery order.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar surat jalan yang baru
Alternatif Flow	-
Post-condition	Data surat jalan telah diperbaharui oleh sistem

Tabel 4.36 Use Case Scenario menghapus surat jalan

Flow of Event untuk menghapus surat jalan	
Kode Kebutuhan	SAT_F_3400
Objective	Use case ini berguna untuk menghapus surat jalan
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu delivery order.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan daftar surat jalan yang baru
Alternatif Flow	-
Post-condition	Sistem akan menghapus order penjualan yang dipilih dari database

Tabel 4.37 Use Case Scenario mencetak surat jalan

Flow of Event untuk mencetak surat jalan	
Kode Kebutuhan	SAT_F_3500
Objective	Use case ini berguna untuk mencetak surat jalan
Actor	Sales Counter

<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu delivery order.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon print pada tabel 2. Sistem akan menampilkan halaman popup 3. Sistem akan menampilkan daftar surat jalan yang akan dicetak dalam bentuk pdf
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Detail daftar surat jalan dalam bentuk pdf

Tabel 4.38 Use Case Scenario membuat surat faktur pajak

<i>Flow of Event untuk membuat surat faktur pajak</i>	
Kode Kebutuhan	SAT_F_3600
<i>Objective</i>	Use case ini berguna untuk membuat surat faktur pajak
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu faktur pajak.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ surat faktur pajak" 2. Sistem menampilkan halaman tambah faktur pajak 3. Aktor memasukkan nomer "Order Penjualan", lalu mengklik tombol "submit" 4. Sistem menampilkan hasil
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data faktur pajak telah tersimpan oleh sistem

Tabel 4.39 Use Case Scenario melihat surat faktur pajak

<i>Flow of Event untuk melihat surat faktur pajak</i>	
Kode Kebutuhan	SAT_F_3800
<i>Objective</i>	Use case ini berguna untuk melihat surat faktur pajak
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter .
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu faktur pajak 2. Sistem menampilkan surat faktur pajak pada tabel
<i>Alternatif Flow</i>	Jika data pada surat faktur pajak tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data faktur pajak ditampilkan dalam tabel

Tabel 4.40 Use Case Scenario mengubah surat faktur pajak

<i>Flow of Event untuk mengubah surat faktur pajak</i>	
Kode Kebutuhan	SAT_F_3700
<i>Objective</i>	Use case ini berguna untuk mengubah surat faktur pajak
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu faktur pajak.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar surat faktur pajak

<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data faktur pajak telah diperbaharui oleh sistem

Tabel 4.41 Use Case Scenario menghapus surat faktur pajak

<i>Flow of Event untuk menghapus surat faktur pajak</i>	
Kode Kebutuhan	SAT_F_3900
<i>Objective</i>	Use case ini berguna untuk menghapus surat faktur pajak
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu faktur pajak.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan daftar surat faktur pajak yang baru
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus surat faktur pajak yang dipilih dari database

Tabel 4.42 Use Case Scenario mencetak surat faktur pajak

<i>Flow of Event untuk mencetak surat faktur pajak</i>	
Kode Kebutuhan	SAT_F_4000
<i>Objective</i>	Use case ini berguna untuk mencetak surat faktur pajak
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu faktur pajak.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon print pada tabel 2. Sistem akan menampilkan halaman popup 3. Sistem akan menampilkan daftar surat faktur pajak yang akan dicetak dalam bentuk pdf
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Detail daftar surat faktur pajak dalam bentuk pdf

Tabel 4.43 Use Case Scenario membuat invoice

<i>Flow of Event untuk membuat invoice</i>	
Kode Kebutuhan	SAT_F_4100
<i>Objective</i>	Use case ini berguna untuk membuat invoice
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu invoice.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ surat faktur pajak" 2. Sistem menampilkan modal 3. Aktor memasukan nomer "Order Penjualan", lalu mengklik tombol "submit" 4. Sistem menampilkan hasil
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data invoice telah tersimpan oleh sistem

Tabel 4.44 Use Case Scenario melihat invoice

<i>Flow of Event untuk melihat invoice</i>	
Kode Kebutuhan	SAT_F_4300
Objective	Use case ini berguna untuk melihat invoice
Actor	Sales counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter.
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu invoice 2. Sistem menampilkan invoice pada tabel
Alternatif Flow	Jika data pada invoice tidak ada, maka akan ditampilkan "tidak ada data"
Post-condition	Data invoice ditampilkan dalam tabel

Tabel 4.45 Use Case Scenario mengubah invoice

<i>Flow of Event untuk mengubah invoice</i>	
Kode Kebutuhan	SAT_F_4200
Objective	Use case ini berguna untuk mengubah invoice
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu invoice.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar invoice
Alternatif Flow	-
Post-condition	Data invoice telah diperbaharui oleh sistem

Tabel 4.46 Use Case Scenario menghapus invoice

<i>Flow of Event untuk menghapus invoice</i>	
Kode Kebutuhan	SAT_F_4400
Objective	Use case ini berguna untuk menghapus invoice
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu invoice.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan daftar invoice yang baru
Alternatif Flow	-
Post-condition	Sistem akan menghapus surat faktur pajak yang dipilih dari database

Tabel 4.47 Use Case Scenario mencetak invoice

<i>Flow of Event untuk mencetak invoice</i>	
Kode Kebutuhan	SAT_F_4600
Objective	Use case ini berguna untuk mencetak invoice
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu invoice.

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon print pada tabel 2. Sistem akan menampilkan halaman popup 3. Sistem akan menampilkan daftar invoice yang akan dicetak dalam bentuk pdf
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Detail invoice dalam bentuk pdf

Tabel 4.48 Use Case Scenario menambah sales order pada invoice

<i>Flow of Event untuk menambah sales order pada invoice</i>	
Kode Kebutuhan	SAT_F_4500
<i>Objective</i>	Use case ini berguna untuk menambah sales order pada invoice
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu invoice.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+ tambah SO" 2. Sistem menampilkan halaman daftar sales order yang ada pada invoice 3. Aktor mengklik tombol "+ tambah SO" 4. Aktor memasukkan nomer sales order, lalu mengklik tombol "submit". 5. Sistem menampilkan daftar sales order yang ada pada invoice.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data sales order telah tersimpan oleh sistem

Tabel 4.49 Use Case Scenario membuat master data pelanggan

<i>Flow of Event untuk membuat master data pelanggan</i>	
Kode Kebutuhan	SAT_F_4700
<i>Objective</i>	Use case ini berguna untuk master data pelanggan
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu master data pelanggan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "+master data pelanggan" 2. Sistem menampilkan form. 3. Aktor memasukkan data data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan daftar master data pelanggan yang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data pelanggan telah tersimpan oleh sistem

Tabel 4.50 Use Case Scenario melihat master data pelanggan

<i>Flow of Event untuk melihat master data pelanggan</i>	
Kode Kebutuhan	SAT_F_4900
<i>Objective</i>	Use case ini berguna untuk melihat master data pelanggan
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu master data pelanggan 2. Sistem menampilkan master data pelanggan pada tabel

<i>Alternatif Flow</i>	Jika data pada master data pelanggan tidak ada, maka akan ditampilkan “tidak ada data”
<i>Post-condition</i>	Data pelanggan ditampilkan pada tabel

Tabel 4.51 Use Case Scenario mengubah master data pelanggan

<i>Flow of Event untuk mengubah master data pelanggan</i>	
Kode Kebutuhan	SAT_F_4800
<i>Objective</i>	Use case ini berguna untuk mengubah master data pelanggan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu master data pelanggan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol “Edit” 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol “submit” 4. Sistem akan menampilkan daftar master data pelanggan
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data pelanggan telah diperbaharui oleh sistem

Tabel 4.52 Use Case Scenario menghapus master data pelanggan

<i>Flow of Event untuk menghapus master data pelanggan</i>	
Kode Kebutuhan	SAT_F_5000
<i>Objective</i>	Use case ini berguna untuk menghapus master data pelanggan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu master data pelanggan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol “Ya” 4. Sistem akan menampilkan master data pelanggan yang baru
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus master data pelanggan yang dipilih dari database

Tabel 4.53 Use Case Scenario membuat master data leasing

<i>Flow of Event untuk membuat master data leasing</i>	
Kode Kebutuhan	SAT_F_5100
<i>Objective</i>	Use case ini berguna untuk master data leasing
<i>Actor</i>	Sales counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu master data leasing.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol “+master data leasing” 2. Sistem menampilkan form. 3. Aktor memasukan data data yang diperlukan, lalu mengklik tombol “submit” 4. Sistem menampilkan daftar master data leasing yang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data leasing telah tersimpan oleh sistem

Tabel 4.54 Use Case Scenario melihat master data leasing

<i>Flow of Event untuk melihat master data leasing</i>	
Kode Kebutuhan	SAT_F_5300
Objective	Use case ini berguna untuk melihat master data leasing
Actor	Sales counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu master data leasing 2. Sistem menampilkan master data leasing pada tabel
Alternatif Flow	Jika data pada master data leasing tidak ada, maka akan ditampilkan "tidak ada data"
Post-condition	Data leasing ditampilkan pada tabel

Tabel 4.55 Use Case Scenario mengubah master data leasing

<i>Flow of Event untuk mengubah master data leasing</i>	
Kode Kebutuhan	SAT_F_5200
Objective	Use case ini berguna untuk mengubah master data leasing
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu master data leasing.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar master data leasing
Alternatif Flow	-
Post-condition	Data leasing telah diperbaharui oleh sistem

Tabel 4.56 Use Case Scenario menghapus master data leasing

<i>Flow of Event untuk menghapus master data leasing</i>	
Kode Kebutuhan	SAT_F_5400
Objective	Use case ini berguna untuk menghapus master data leasing
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu master data leasing.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan master data leasing yang baru
Alternatif Flow	-
Post-condition	Sistem akan menghapus master data leasing yang dipilih dari database

Tabel 4.57 Use Case Scenario membuat data kas master

<i>Flow of Event untuk membuat data kas master</i>	
Kode Kebutuhan	SAT_F_5500
Objective	Use case ini berguna untuk data kas master
Actor	Kasir

<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas master.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Add New" 2. Sistem menampilkan form. 3. Aktor memasukkan data data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan daftar data kas master yang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data kas master telah tersimpan oleh sistem

Tabel 4.58 Use Case Scenario melihat data kas master

<i>Flow of Event untuk melihat data kas master</i>	
Kode Kebutuhan	SAT_F_5700
<i>Objective</i>	Use case ini berguna untuk melihat data kas master
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu kas master 2. Sistem menampilkan data kas master pada tabel
<i>Alternatif Flow</i>	Jika data pada data kas master tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data kas master ditampilkan pada tabel

Tabel 4.59 Use Case Scenario mengubah data kas master

<i>Flow of Event untuk mengubah data kas master</i>	
Kode Kebutuhan	SAT_F_5600
<i>Objective</i>	Use case ini berguna untuk mengubah data kas master
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas master.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar data kas master
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data kas master telah diperbaharui oleh sistem

Tabel 4.60 Use Case Scenario menghapus data kas master

<i>Flow of Event untuk menghapus data kas master</i>	
Kode Kebutuhan	SAT_F_5800
<i>Objective</i>	Use case ini berguna untuk menghapus data kas master
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas master.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan data kas master yang baru
<i>Alternatif Flow</i>	-

<i>Post-condition</i>	Sistem akan menghapus data kas master yang dipilih dari database
-----------------------	------------------------------------------------------------------

Tabel 4.61 Use Case Scenario membuat data kas terima

<i>Flow of Event untuk</i> membuat data kas terima	
Kode Kebutuhan	SAT_F_5900
<i>Objective</i>	Use case ini berguna untuk data kas terima
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas terima.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Add New" 2. Sistem menampilkan form. 3. Aktor memasukan data data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan daftar data kas terima yang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data kas terima telah tersimpan oleh sistem

Tabel 4.62 Use Case Scenario melihat data kas terima

<i>Flow of Event untuk</i> melihat data kas terima	
Kode Kebutuhan	SAT_F_6100
<i>Objective</i>	Use case ini berguna untuk melihat data kas terima
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu kas terima 2. Sistem menampilkan data kas terima pada tabel
<i>Alternatif Flow</i>	Jika data pada data kas terima tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data kas terima ditampilkan pada tabel

Tabel 4.63 Use Case Scenario mengubah data kas terima

<i>Flow of Event untuk</i> mengubah data kas terima	
Kode Kebutuhan	SAT_F_6000
<i>Objective</i>	Use case ini berguna untuk mengubah data kas terima
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas terima.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar data kas terima
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data kas terima telah diperbaharui oleh sistem

Tabel 4.64 Use Case Scenario menghapus data kas terima

<i>Flow of Event untuk</i> menghapus data kas terima	
Kode Kebutuhan	SAT_F_6200
<i>Objective</i>	Use case ini berguna untuk menghapus data kas terima

<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas terima.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan data kas terima yang baru
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus data kas terima yang dipilih dari database

Tabel 4.65 Use Case Scenario membuat data kas keluar

<i>Flow of Event untuk membuat data kas keluar</i>	
Kode Kebutuhan	SAT_F_6300
<i>Objective</i>	Use case ini berguna untuk data kas keluar
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas keluar.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Add New" 2. Sistem menampilkan form. 3. Aktor memasukkan data data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan daftar data kas keluar yang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data kas keluar telah tersimpan oleh sistem

Tabel 4.66 Use Case Scenario melihat data kas keluar

<i>Flow of Event untuk melihat data kas keluar</i>	
Kode Kebutuhan	SAT_F_6500
<i>Objective</i>	Use case ini berguna untuk melihat data kas keluar
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu kas keluar 2. Sistem menampilkan data kas keluar pada tabel
<i>Alternatif Flow</i>	Jika data pada data kas keluar tidak ada, maka akan ditampilkan "tidak ada data"
<i>Post-condition</i>	Data kas keluar ditampilkan pada tabel

Tabel 4.67 Use Case Scenario mengubah data kas keluar

<i>Flow of Event untuk mengubah data kas keluar</i>	
Kode Kebutuhan	SAT_F_6400
<i>Objective</i>	Use case ini berguna untuk mengubah data kas keluar
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas keluar.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Edit" 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol "submit" 4. Sistem akan menampilkan daftar data kas keluar

<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data kas keluar telah diperbaharui oleh sistem

Tabel 4.68 Use Case Scenario menghapus data kas keluar

<i>Flow of Event untuk</i> menghapus data kas keluar	
Kode Kebutuhan	SAT_F_6600
<i>Objective</i>	Use case ini berguna untuk menghapus data kas keluar
<i>Actor</i>	Kasir
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai kasir dan telah memilih menu kas keluar.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan data kas keluar yang baru
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Sistem akan menghapus data kas terima yang dipilih dari database

Tabel 4.69 Use Case Scenario melihat riwayat kas terima

<i>Flow of Event untuk</i> melihat riwayat kas terima	
Kode Kebutuhan	SAT_F_6700
<i>Objective</i>	Use case ini berguna untuk melihat riwayat kas terima
<i>Actor</i>	Manager
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai manager
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu riwayat kas terima 2. Sistem menampilkan data riwayat kas terima pada tabel
<i>Alternatif Flow</i>	Jika data pada riwayat kas terima tidak ada, maka akan tampil "tidak ada data"
<i>Post-condition</i>	Data kas terima ditampilkan pada tabel

Tabel 4.70 Use Case Scenario melihat riwayat kas keluar

<i>Flow of Event untuk</i> melihat riwayat kas keluar	
Kode Kebutuhan	SAT_F_6800
<i>Objective</i>	Use case ini berguna untuk melihat riwayat kas keluar
<i>Actor</i>	Manager
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai manager.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu riwayat kas keluar 2. Sistem menampilkan data riwayat kas keluar pada tabel
<i>Alternatif Flow</i>	Jika data pada riwayat kas keluar tidak ada, maka akan tampil "tidak ada data"
<i>Post-condition</i>	Data kas keluar ditampilkan pada tabel

Tabel 4.71 Use Case Scenario melihat riwayat sales order

<i>Flow of Event untuk</i> melihat riwayat sales order	
Kode Kebutuhan	SAT_F_6900
<i>Objective</i>	Use case ini berguna untuk melihat riwayat sales order
<i>Actor</i>	Manager
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai manager.

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu riwayat sales order 2. Sistem menampilkan data riwayat sales order pada tabel
<i>Alternatif Flow</i>	Jika data pada riwayat sales order tidak ada, maka akan tampil “tidak ada data”
<i>Post-condition</i>	Data sales order ditampilkan pada tabel

Tabel 4.72 Use Case Scenario membuat data antrian servis

<i>Flow of Event untuk membuat data antrian servis</i>	
Kode Kebutuhan	SAT_F_7000
<i>Objective</i>	Use case ini berguna untuk data antrian servis
<i>Actor</i>	Mekanik
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai mekanik dan telah memilih menu daftar antrian servis.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol “Add New” 2. Sistem menampilkan form. 3. Aktor memasukkan data data yang diperlukan, lalu mengklik tombol “submit” 4. Sistem menampilkan daftar data antrian servis yang baru.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data antrian servis telah tersimpan oleh sistem

Tabel 4.73 Use Case Scenario melihat data antrian servis

<i>Flow of Event untuk melihat data antrian servis</i>	
Kode Kebutuhan	SAT_F_7200
<i>Objective</i>	Use case ini berguna untuk melihat data antrian servis
<i>Actor</i>	Mekanik
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai mekanik.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu daftar antrian servis 2. Sistem menampilkan data antrian servis pada tabel
<i>Alternatif Flow</i>	Jika data pada data antrian servis tidak ada, maka akan ditampilkan “tidak ada data”
<i>Post-condition</i>	Data antrian servis ditampilkan pada tabel

Tabel 4.74 Use Case Scenario mengubah data antrian servis

<i>Flow of Event untuk mengubah data antrian servis</i>	
Kode Kebutuhan	SAT_F_7100
<i>Objective</i>	Use case ini berguna untuk mengubah data antrian servis
<i>Actor</i>	Mekanik
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai mekanik dan telah memilih menu daftar antrian servis.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol “Edit” 2. Sistem akan menampilkan halaman edit item. 3. Aktor akan mengisi input dari data yang akan dirubah, setelah itu mengklik tombol “submit” 4. Sistem akan menampilkan daftar data antrian servis
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Data antrian telah diperbaharui oleh sistem

Tabel 4.75 Use Case Scenario menghapus data antrian servis

<i>Flow of Event untuk</i> menghapus data antrian servis	
Kode Kebutuhan	SAT_F_7300
Objective	Use case ini berguna untuk menghapus data antrian servis
Actor	Mekanik
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai mekanik dan telah memilih menu daftar antrian servis.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol ikon sampah pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan data antrian servis yang baru
Alternatif Flow	-
Post-condition	Sistem akan menghapus data antrian servis yang dipilih dari database

Tabel 4.76 Use Case Scenario merubah status antrian servis

<i>Flow of Event untuk</i> merubah status antrian servis	
Kode Kebutuhan	SAT_F_7400
Objective	Use case ini berguna untuk merubah status antrian servis
Actor	Mekanik
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai mekanik dan telah memilih menu daftar antrian servis.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Selesai" pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya" 4. Sistem akan menampilkan data antrian servis yang baru
Alternatif Flow	-
Post-condition	Sistem akan merubah status antrian servis yang dipilih dari database

Tabel 4.77 Use Case Scenario melakukan pengecekan gratis servis

<i>Flow of Event untuk</i> melakukan pengecekan gratis servis	
Kode Kebutuhan	SAT_F_7500
Objective	Use case ini berguna untuk melakukan pengecekan gratis servis
Actor	Mekanik
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai mekanik dan telah memilih menu cek gratis servis.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengisi data data yang diperlukan pada form, lalu mengklik tombol "submit"
Alternatif Flow	Jika data motor tidak ada dalam database, maka sistem akan menampilkan data "Data motor tidak ada dalam database"
Post-condition	Sistem akan menampilkan informasi yang berisi apakah motor tersebut masih berlaku masa gratis servisnya

Tabel 4.78 Use Case Scenario melihat riwayat pesan kepada pelanggan

<i>Flow of Event untuk</i> melihat riwayat pesan kepada pelanggan	
Kode Kebutuhan	SAT_F_7600
Objective	Use case ini berguna untuk melihat riwayat pesan kepada pelanggan
Actor	Sales Counter

<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu layanan pelanggan 2. Sistem menampilkan riwayat pesan pelanggan
<i>Alternatif Flow</i>	Jika data riwayat tidak ada dalam database, maka sistem akan menampilkan data "Data riwayat pesan tidak ada dalam database"
<i>Post-condition</i>	Data riwayat akan ditampilkan pada tabel

Tabel 4.79 Use Case Scenario Mengirim pesan kepada pelanggan

<i>Flow of Event untuk Mengirim pesan kepada pelanggan</i>	
Kode Kebutuhan	SAT_F_7700
<i>Objective</i>	Use case ini berguna untuk Mengirim pesan kepada pelanggan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu layanan pelanggan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Add New" 2. Sistem menampilkan form. 3. Aktor memasukkan data data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan status pesan pada tabel riwayat
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Pesan akan diproses untuk dikirim kepada pelanggan

Tabel 4.80 Use Case Scenario Mengirim pesan kepada banyak pelanggan

<i>Flow of Event untuk Mengirim pesan kepada banyak pelanggan</i>	
Kode Kebutuhan	SAT_F_7800
<i>Objective</i>	Use case ini berguna untuk Mengirim pesan kepada banyak pelanggan
<i>Actor</i>	Sales Counter
<i>Pre-condition</i>	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu layanan pelanggan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Add New" 2. Sistem menampilkan form. 3. Aktor memasukkan data data yang diperlukan, lalu mengklik tombol "submit" 4. Sistem menampilkan status pesan pada tabel riwayat
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Pesan akan diproses untuk dikirim kepada pelanggan

Tabel 4.81 Use Case Scenario Mengirim pesan otomatis pengingat servis

<i>Flow of Event untuk Mengirim pesan otomatis pengingat servis</i>	
Kode Kebutuhan	SAT_F_7900
<i>Objective</i>	Use case ini berguna untuk Mengirim pesan otomatis pengingat servis
<i>Actor</i>	CronJob
<i>Pre-condition</i>	Cronjon berjalan setiap hari jam 7 pagi
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem melakukan pengecekan data sales order 2. Sistem mengirim pengingat kepada pelanggan
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Pesan pengingat akan diproses untuk dikirim kepada pelanggan

Tabel 4.82 Use Case Scenario Mengirim pesan otomatis ucapan ulang tahun

<i>Flow of Event untuk Mengirim pesan otomatis ucapan ulang tahun</i>	
Kode Kebutuhan	SAT_F_8000
Objective	Use case ini berguna untuk Mengirim pesan otomatis ucapan ulang tahun
Actor	CronJob
Pre-condition	Cronjon berjalan setiap hari jam 7 pagi
Main Flow	<ol style="list-style-type: none"> 1. Sistem melakukan pengecekan data pribadi pelanggan 2. Sistem mengirim ucapan ulang tahun kepada pelanggan
Alternatif Flow	-
Post-condition	Pesan pengingat akan diproses untuk dikirim kepada pelanggan

Tabel 4.83 Use Case Scenario mengunci status invoice

<i>Flow of Event untuk mengunci status bukti penerimaan barang</i>	
Kode Kebutuhan	SAT_F_8100
Objective	Use case ini berguna untuk mengunci status invoice
Actor	Sales Counter
Pre-condition	Aktor sudah masuk ke dalam sistem dengan hak akses sebagai sales counter dan telah memilih menu invoice.
Main Flow	<ol style="list-style-type: none"> 1. Aktor mengklik tombol "Lock" pada tabel 2. Sistem akan menampilkan modal konfirmasi 3. Aktor mengklik tombol "Ya"
Alternatif Flow	-
Post-condition	Status invoice akan menjadi terkunci dan tidak dapat dirubah

4.6 Pembahasan Hasil Analisis Kebutuhan

Dari analisis kebutuhan yang dilakukan, sistem ini akan menghasilkan 81 kebutuhan yang dapat membantu pekerjaan karyawan yang ada pada PT. Mega Abadi Motorindo. Masing masing pengguna, baik dari admin, sales counter, kasir, manager dan mekanik akan mendapatkan hak akses sesuai dengan kebutuhan yang telah didefinisikan sebelumnya.

BAB 5 PERANCANGAN DAN IMPLEMENTASI SISTEM

5.1 Perancangan sistem

Setelah proses analisis kebutuhan selesai dilakukan, tahap berikutnya adalah perancangan. Perancangan dilakukan berdasarkan hasil dari analisis kebutuhan yang telah dilakukan. Proses perancangan aplikasi pada PT. Mega Abadi Motorindo ini dibagi menjadi beberapa tahap yaitu melakukan pemodelan sequence diagram, pemodelan class diagram, pemodelan entity relationship diagram, dan perancangan antar muka.

5.1.1 Pemodelan Sequence Diagram

Pada pemodelan sequence diagram menjelaskan urutan proses yang terjadi untuk mencapai suatu kebutuhan sistem. Objek-objek pada sequence diagram merupakan hasil identifikasi dari spesifikasi kebutuhan dan *Use Case Scenario* yang ada pada tahap analisis kebutuhan. Setiap sequence diagram menggambarkan setiap use case yang ada. Pemodelan sequence diagram tersebut dijelaskan pada sub-sub bab dibawah ini.

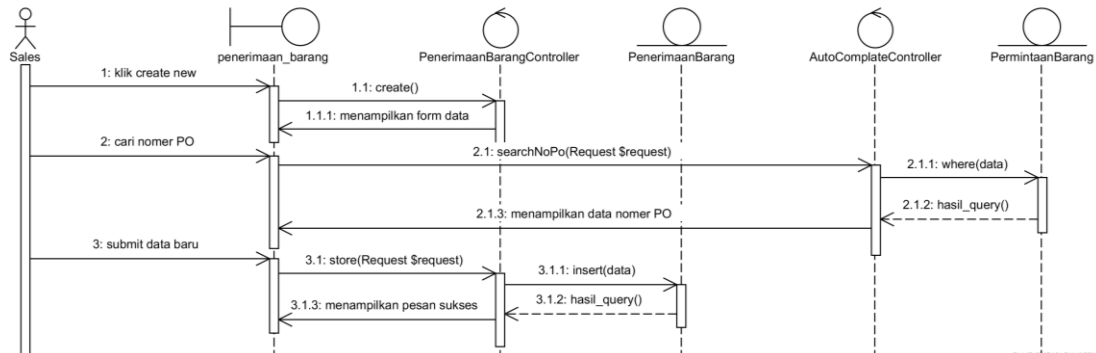
5.1.1.1 Sequence Diagram Membuat order penjualan baru



Gambar 5.1 Sequence Diagram Membuat order penjualan baru

Dalam gambar 5.1 *sequence diagram* membuat order penjualan baru, sales dapat menginputkan data yang diperlukan seperti nama pelanggan, nama tagihan, barang berasal dari gudang internal atau eksternal, harga barang termasuk cash, cash tempo atau credit, dan menggunakan leasing atau tidak. Setelah sales mengirimkan data, *SalesOrderController* akan memanggil method *store* dengan parameter *Request* untuk melakukan pengolahan data. Setelah pengolahan data selesai maka data akan dikirimkan ke model *SalesOrder* untuk dilakukan *insert* data pada database.

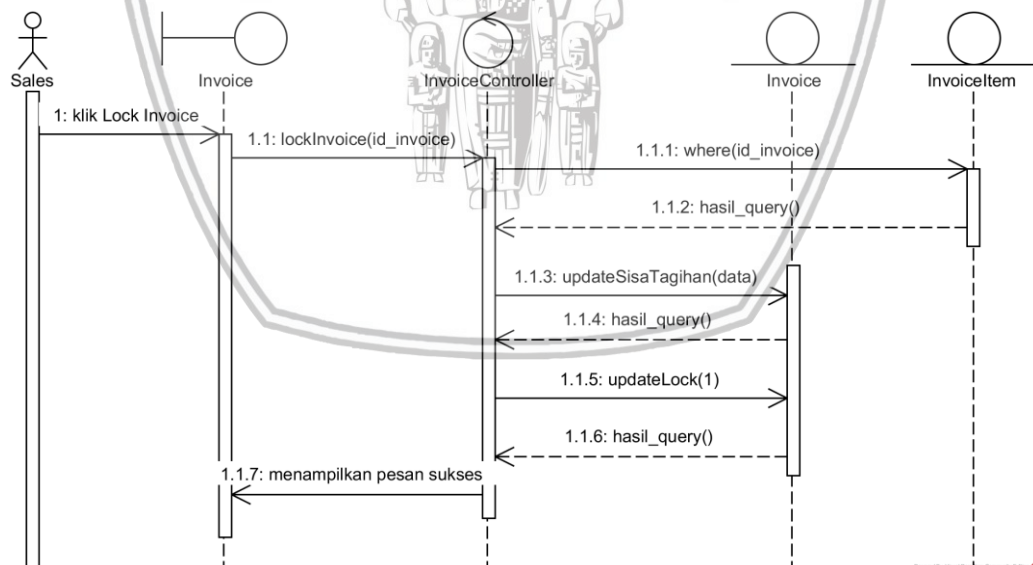
5.1.1.2 Sequence Diagram Membuat bukti penerimaan barang



Gambar 5.2 Sequence Diagram Membuat bukti penerimaan barang

Dalam gambar 5.2 *sequence diagram* membuat bukti penerimaan barang, sales dapat menginputkan data yang diperlukan seperti Nomer PO, No Ref dan Data Gudang. Pada saat memasukan nomer PO, view penerimaan barang akan memanggil fungsi pada AutoComplateController untuk mencari data nomer PO yang disedang diinputkan oleh sales. Setelah sales melakukan submit data, method store pada PenerimaanBarangController akan dijalankan. Setelah pengolahan data selesai maka data akan dikirimkan ke model SalesOrder untuk dilakukan insert data pada database.

5.1.1.3 Sequence Diagram mengunci status invoice



Gambar 5.3 Sequence Diagram Lock Invoice

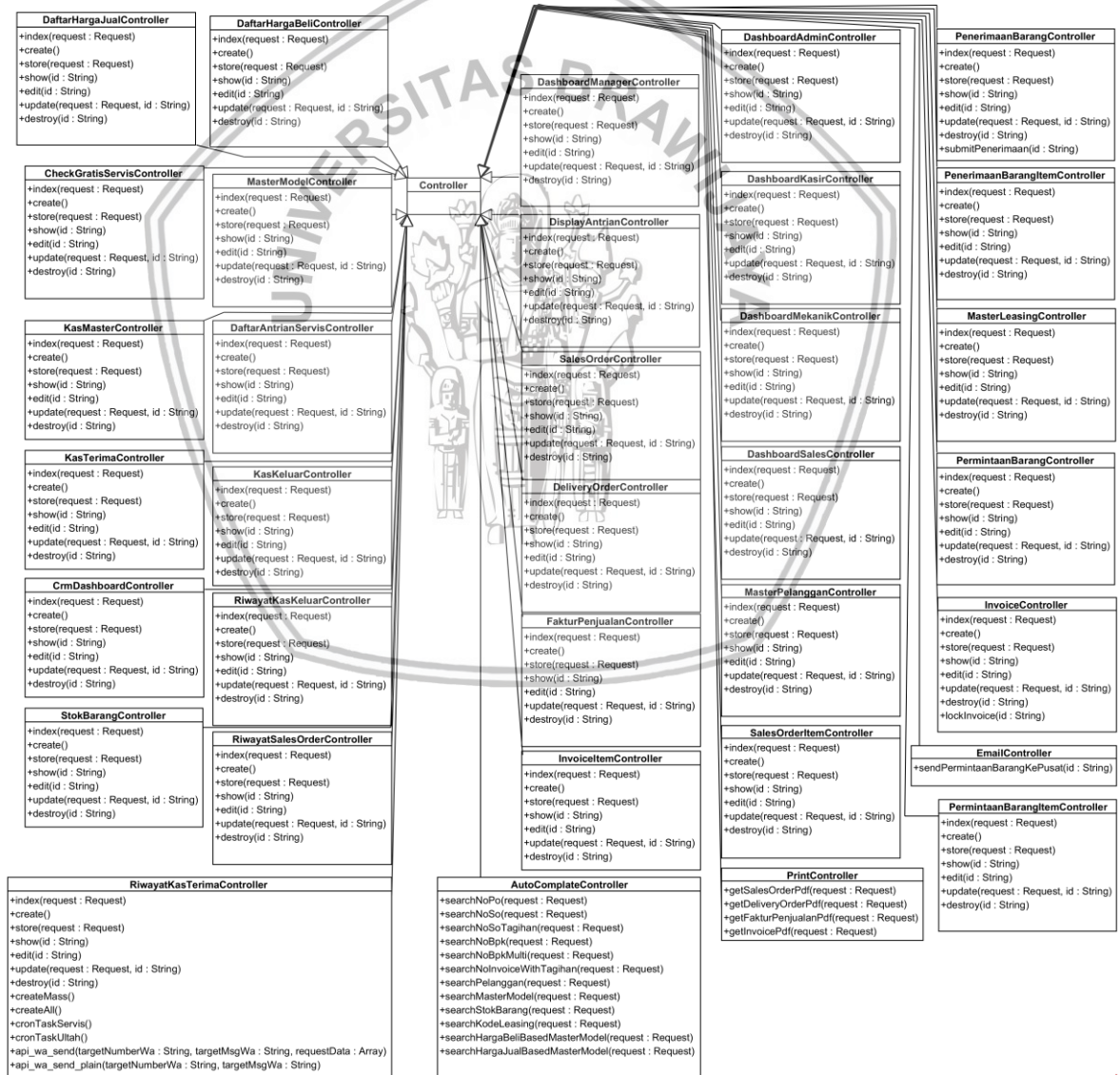
Dalam gambar 5.3 *sequence diagram* lock invoice, saat sales menekan tombol "lock invoice" maka akan memanggil method lockInvoice dengan parameter id_invoice pada InvoiceController. Sebelum melakukan lock invoice, InvoiceController akan mencari data InvoiceItem yang sesuai dengan id invoice yang sedang di proses. Setelah mendapatkan hasil dari model InvoiceItem,

InvoiceController akan melakukan query pada model Invoice untuk melakukan updateTagihan dan merubah status lock menjadi bernilai satu (1).

5.1.2 Pemodelan Class Diagram

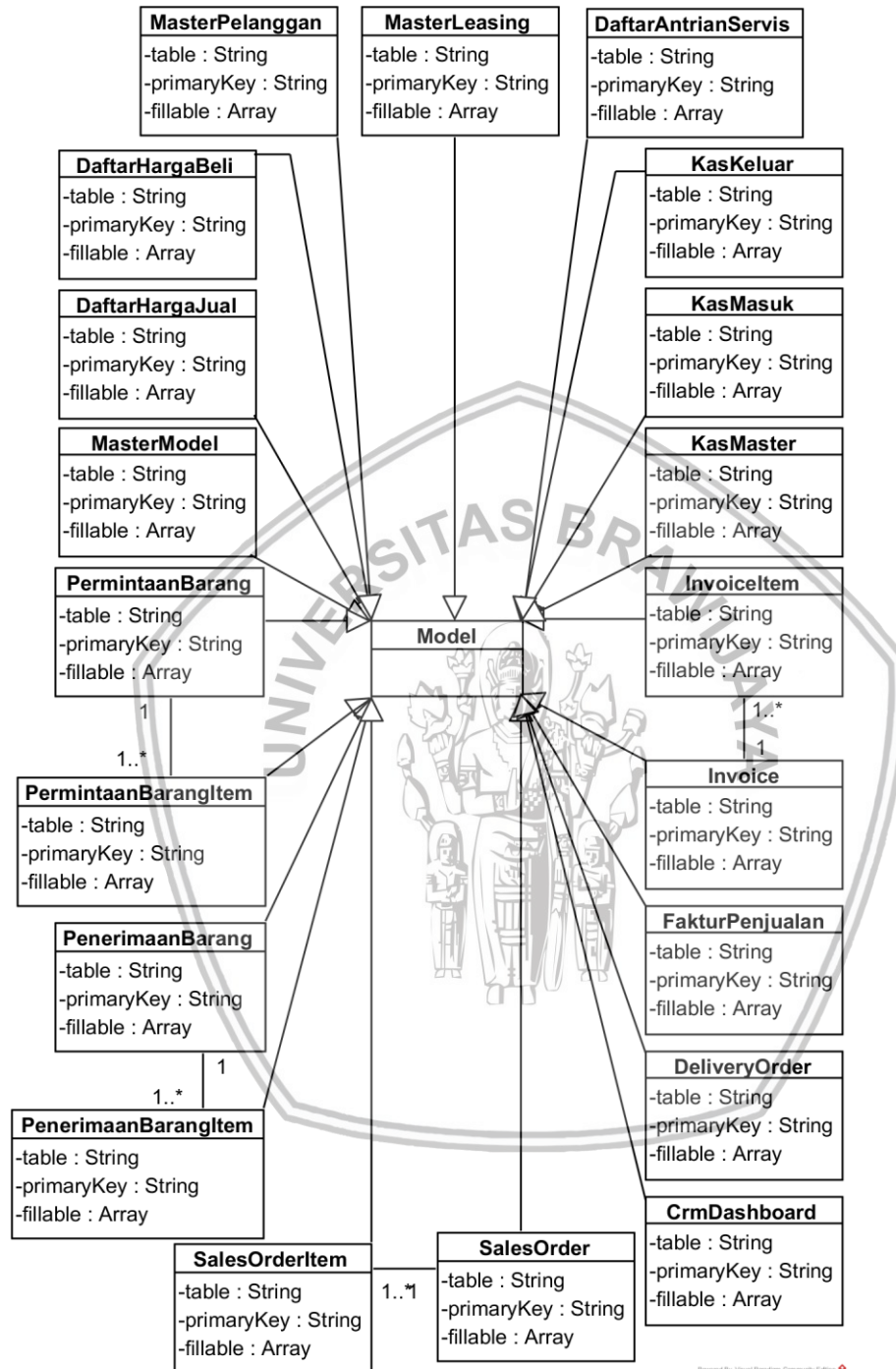
Pemodelan *class diagram* dilakukan untuk menggambarkan objek-objek yang terbentuk dan relasi antara objek objek tersebut. Pemodelan *class diagram* ini akan menjelaskan pada controller dan model.

Class diagram pada controller mempunyai 33 kelas yang meng-extends klas *Controller*. Pada kelas *AutoComplateController* digunakan untuk melakukan proses data saat user melakukan *lookup* data pada form. Pada *AutoComplateController* ini nilai balikan datanya berbeda dengan yang lain, yakni mengembalikan tipe data json. Pemodelan class diagram pada controller digambarkan pada gambar 5.4.



Gambar 5.4 Class Diagram pada Controller

Class diagram pada model mempunyai 19 kelas yang meng-extends kelas Model. Pemodelan class diagram pada model digambarkan pada gambar 5.5



Gambar 5.5 Class Diagram pada Model

5.1.3 Perancangan Komponen

Perancangan komponen menggambarkan rincian sub-sistem dari setiap komponen perangkat lunak. Untuk mencapai hal ini, perancangan komponen harus mendefinisikan struktur data untuk semua objek data local dan rincian algoritme untuk proses yang terjadi di dalam komponen. Dalam perancangan komponen ini dipilih satu algoritme dari setiap kelas pada *controller*.

5.1.3.1 Perancangan Komponen Klas EmailController

Nama Operasi : sendPermintaanBarangKePusat()

Algoritme :

```
inisialisasi variabel permintaanbarang = melakukan query database pada
tabel permintaanbarang
inisialisasi variabel permintaanbarangitem = melakukan query database
pada tabel permintaanbarangitem
melakukan query database pada tabel permintaanbarang untuk merubah status
permintaan barang menjadi telah terkirim
memanggil fungsi send() pada class Mail dengan menggunakan parameter
variabel permintaanbarang, variabel permintaanbarangitem dan email
tujuan.
If (count(Mail::failures()) > 0)
    Inisialisasi variable error = Failed to send password reset email,
    please try again.
Else
    Inisialisasi variable error = email send
mengembalikan variabel error untuk melakukan redirect ke halaman
Permintaan Barang
```

5.1.3.2 Perancangan Komponen Klas AutoCompleteController

Nama Operasi :searchNoPo()

Algoritme :

```
Inisialisasi variabel query = memanggil fungsi get() pada request dengan
parameter term
Melakukan query ke database pada tabel permintaan_barang
Inisialisasi array data
Foreach(products as product)
    Array data = array(value=>product->nopo,id=>product->id)
If(count(data))
    Mengembalikan variabel data
Else
    Menembalikan nilai no result found
```


63

Tabel 5.1 Struktur Tabel master_pelanggan

Nama Kolom	Tipe	Keterangan
id	int(10)	ID pelanggan, digunakan sebagai identifikasi
created_at	date	Tanggal data dibuat
updated_at	date	Tanggal data dirubah
pelangganids	varchar	ID pelanggan yang ditentukan oleh pengguna
pelangganname	varchar	Nama pelanggan
pelangganinisial	varchar	Inisial pelanggan
gender	varchar	Gender pelanggan
nohp	varchar	NoHP pelanggan
notelepon	varchar	NoTelepon pelanggan
alamatemail	varchar	Alamat email pelanggan
tanggallahir	date	Tanggal lahir pelanggan
alamat	varchar	Alamat pelanggan
kodepos	varchar	Kodepos pelanggan
kota	varchar	Kota pelanggan
provinsi	varchar	Provinsi pelanggan
status	varchar	Status pelanggan

Tabel 5.2 Struktur Tabel sales_order

Nama Kolom	Tipe	Keterangan
id	int(10)	ID sales_order, digunakan sebagai identifikasi
created_at	date	Tanggal data dibuat
updated_at	date	Tanggal data dirubah
pelanggan	int(10)	ID pelanggan
salesman	varchar	Salesman yang mengisi data
tagihanpada	varchar	Tagihan sales order
gudang	varchar	Gudang barang yang dijual
groupprice	varchar	Kelompok harga
leasing	varchar	Status leasing
kodeleasing	varchar	Kode leasing yang digunakan

Tabel 5.3 Struktur Tabel faktur_penjualan

Nama Kolom	Tipe	Keterangan
id	int(10)	ID faktur_penjualan, digunakan sebagai identifikasi
created_at	date	Tanggal data dibuat
updated_at	date	Tanggal data dirubah
nobpk	varchar	No BPK yang termasuk dalam faktur
noso	varchar	No SO yang termasuk dalam faktur

Tabel 5.4 Struktur Tabel invoice

Nama Kolom	Tipe	Keterangan
id	int(10)	ID invoice, digunakan sebagai identifikasi
created_at	date	Tanggal data dibuat
updated_at	date	Tanggal data dirubah
noinv	int(10)	No invoice
tagihke	varchar	Tagihan kepada pelanggan
sisatagihan	varchar	Sisa tagihan
keterangan	varchar	Keterangan tambahan
submitlock	int(1)	Status invoice lock

Tabel 5.5 Struktur Tabel kas_master

Nama Kolom	Tipe	Keterangan
id	int(10)	ID pelanggan, digunakan sebagai identifikasi
created_at	date	Tanggal data dibuat
updated_at	date	Tanggal data dirubah
idkas	varchar	Alias dari kas
namakas	varchar	Nama kas
jumlah	varchar	Jumlah dari kas

5.1.5 Perancangan Antarmuka Aplikasi

Dalam perancangan antarmuka ini akan ditunjukkan perancangan layout antarmuka yang akan diimplementasi pada sistem. Dalam perancangan antarmuka ini akan ditunjukkan beberapa sampel antar muka seperti antarmuka list sales order, invoice.

5.1.5.1 Antarmuka *list sales order*

Perancangan antarmuka *list sales order* akan dijelaskan pada Gambar 5.8 ,dan penejelasan detail dari perancangan antar muka tersebut akan dijelaskan pada Tabel 5.6



Gambar 5.8 Perancangan antarmuka sales order

Tabel 5.6 Penjelasan antarmuka sales order

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	Logo button	Menampilkan judul aplikasi berupa logo
2	Nama Pengguna	Menu dropdown	Menu untuk menampilkan username dan logout user
3	Menu Utama	Menu	Menampilkan menu yang dapat diakses oleh user
4	Add Sales Order	Menu	Menu untuk melakukan pembuatan data pada sales order
5	Search Sales Order	Form	Melakukan pencarian data pada sales order
6	No SO	Text	Informasi header tabel untuk No SO
7	Pelanggan	Text	Informasi header tabel untuk Pelanggan
8	Salesman	Text	Informasi header tabel untuk Salesman
9	Action	Text	Informasi header tabel untuk Action yang tersedia
10	View	Button	Menu untuk melihat detail sales order
11	Edit	Button	Menu untuk mengubah data sales order
12	Delete	Button	Menu untuk menghapus data sales order

13	Tambah Item Sepeda	Button	Menu untuk menambah item sepeda motor pada sales order
14	Print SO	Button	Menu untuk mencetak sales order

5.1.5.2 Antarmuka invoice

Perancangan antarmuka list sales order akan dijelaskan pada Gambar 5.9 ,dan penejelasan detail dari perancangan antar muka tersebut akan dijelaskan pada Tabel 5.7



Gambar 5.9 Perancangan antarmuka list invoice

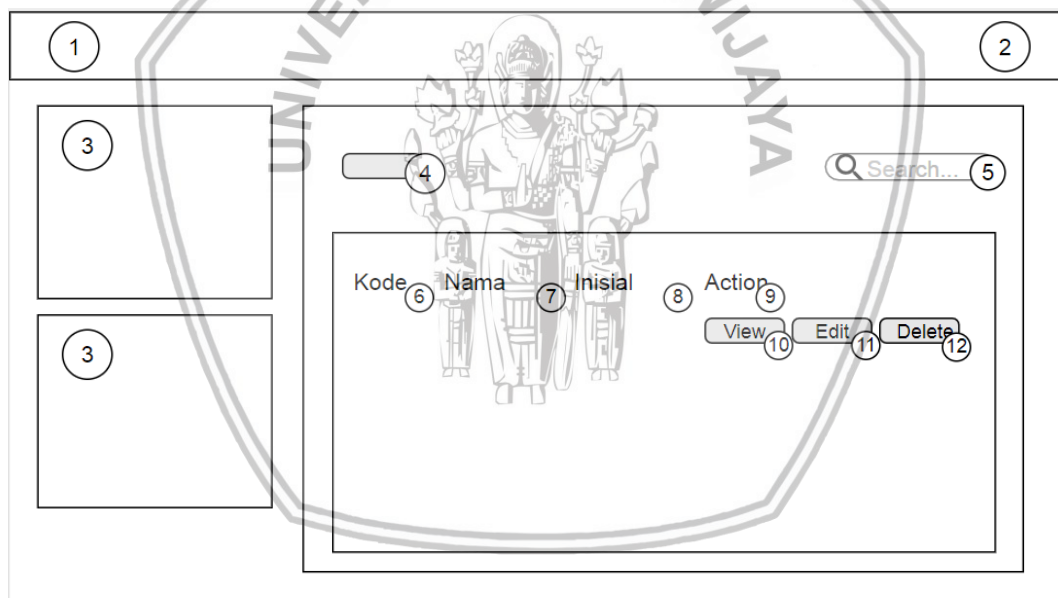
Tabel 5.7 Penjelasan antarmuka list invoice

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	Logo button	Menampilkan judul aplikasi berupa logo
2	Nama Pengguna	Menu dropdown	Menu untuk menampilkan username dan logout user
3	Menu Utama	Menu	Menampilkan menu yang dapat diakses oleh user
4	Add Invoice	Menu	Menu untuk melakukan pembuatan data pada invoice
5	Search	Form	Melakukan pencarian data pada invoice
6	No Inv	Text	Informasi header tabel untuk No INV
7	Tagih ke	Text	Informasi header tabel untuk tagihan ke
8	Sisa Tagihan	Text	Informasi header tabel untuk Sisa Tagihan

9	Action	Text	Informasi header tabel untuk Action yang tersedia
10	View	Button	Menu untuk melihat detail invoice
11	Edit	Button	Menu untuk mengubah data invoice
12	Delete	Button	Menu untuk menghapus data invoice
13	Tambah Sales Order	Button	Menu untuk menambah sales order pada invoice
14	Lock Inv	Button	Menu untuk mengunci data invoice
15	Keterangan	Button	Informasi header tabel untuk Keterangan
16	Print Inv	Button	Menu untuk mencetak invoice

5.1.5.3 Antarmuka master pelanggan

Perancangan antarmuka master pelanggan akan dijelaskan pada Gambar 5.10, dan penjelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.8



Gambar 5.10 Perancangan antarmuka master pelanggan

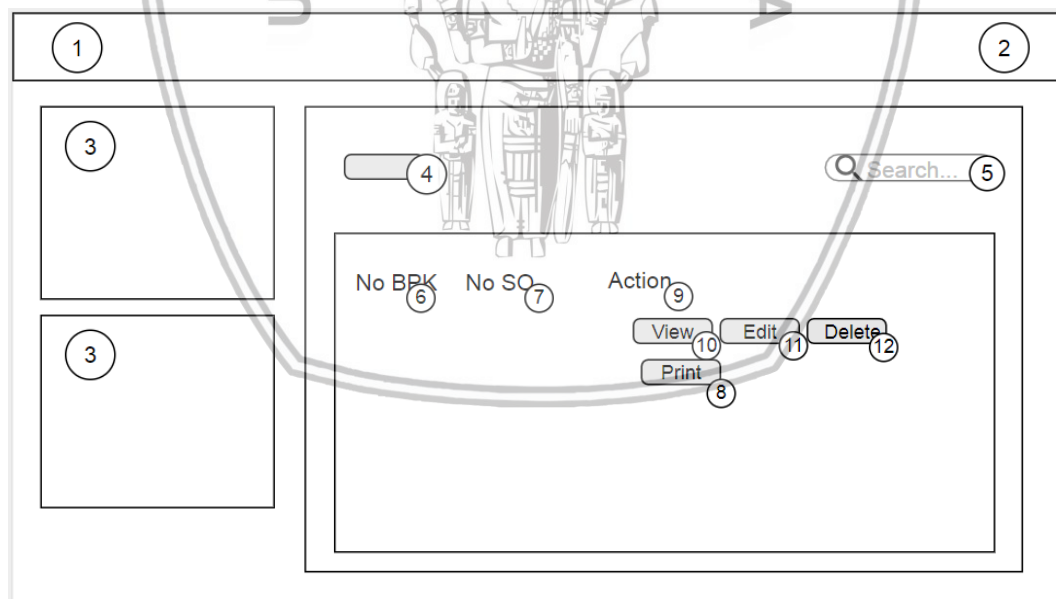
Tabel 5.8 Penjelasan antarmuka master pelanggan

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	Logo button	Menampilkan judul aplikasi berupa logo
2	Nama Pengguna	Menu dropdown	Menu untuk menampilkan username dan logout user
3	Menu Utama	Menu	Menampilkan menu yang dapat diakses oleh user

4	Add	Menu	Menu untuk melakukan pembuatan data pada master pelanggan
5	Search	Form	Melakukan pencarian data pada master pelanggan
6	Kode	Text	Informasi header tabel untuk Kode
7	Nama	Text	Informasi header tabel untuk nama
8	Inisial	Text	Informasi header tabel untuk Inisial
9	Action	Text	Informasi header tabel untuk Action yang tersedia
10	View	Button	Menu untuk melihat informasi detail
11	Edit	Button	Menu untuk mengubah data master pelanggan
12	Delete	Button	Menu untuk menghapus data master pelanggan

5.1.5.4 Antarmuka faktur penjualan

Perancangan antarmuka faktur penjualan akan dijelaskan pada Gambar 5.11 ,dan penejelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.9



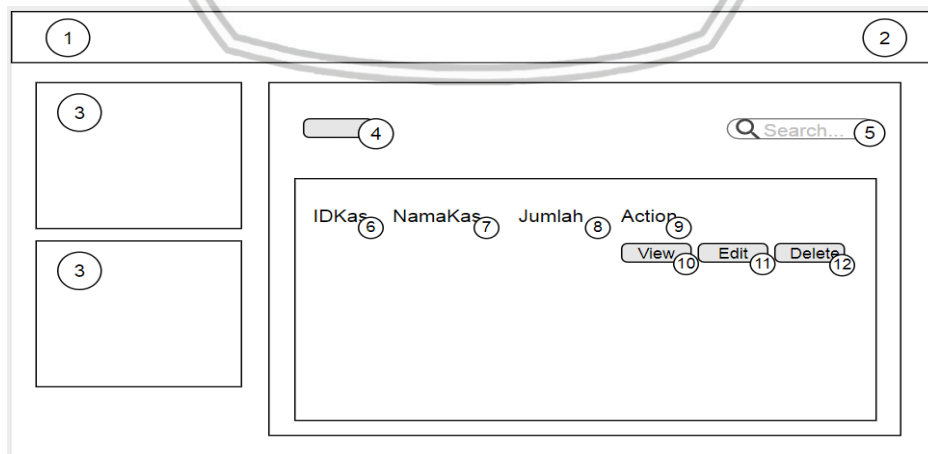
Gambar 5.11 Perancangan antarmuka faktur penjualan

Tabel 5.9 Penjelasan antarmuka master pelanggan

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	Logo button	Menampilkan judul aplikasi berupa logo
2	Nama Pengguna	Menu dropdown	Menu untuk menampilkan username dan logout user
3	Menu Utama	Menu	Menampilkan menu yang dapat diakses oleh user
4	Add	Menu	Menu untuk melakukan pembuatan data pada faktur penjualan
5	Search	Form	Melakukan pencarian data pada faktur penjualan
6	No BPK	Text	Informasi header tabel untuk No BPK
7	No SO	Text	Informasi header tabel untuk No SO
8	Print	Button	Menu untuk mencetak faktur penjualan
9	Action	Text	Informasi header tabel untuk Action yang tersedia
10	View	Button	Menu untuk melihat informasi detail
11	Edit	Button	Menu untuk mengubah data faktur penjualan
12	Delete	Button	Menu untuk menghapus data faktur penjualan

5.1.5.5 Antarmuka kas master

Perancangan antarmuka kas master akan dijelaskan pada Gambar 5.12 ,dan penejelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.7



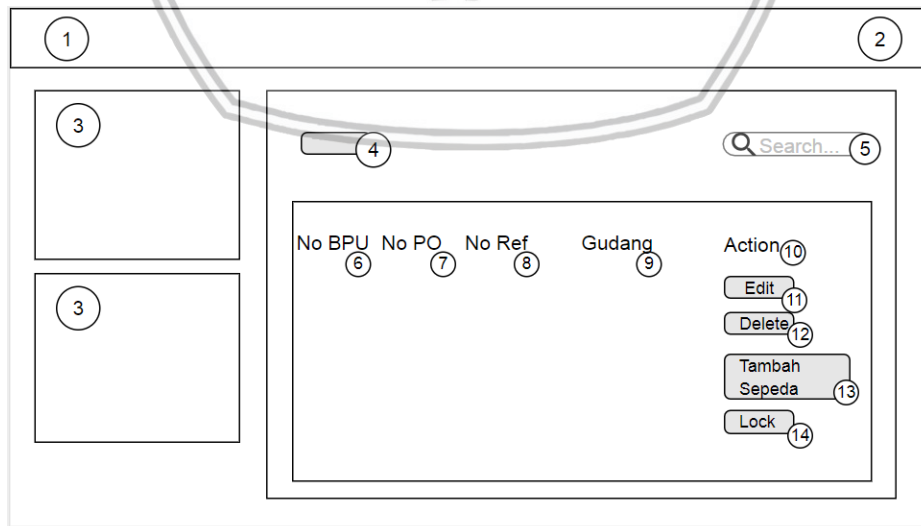
Gambar 5.12 Perancangan antarmuka kas master

Tabel 5.10 Penjelasan antarmuka kas master

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	Logo button	Menampilkan judul aplikasi berupa logo
2	Nama Pengguna	Menu dropdown	Menu untuk menampilkan username dan logout user
3	Menu Utama	Menu	Menampilkan menu yang dapat diakses oleh user
4	Add	Menu	Menu untuk melakukan pembuatan data pada kas master
5	Search	Form	Melakukan pencarian data pada kas master
6	ID kas	Text	Informasi header tabel untuk id kas
7	Nama Kas	Text	Informasi header tabel untuk nama kas
8	Jumlah	Text	Informasi header tabel untuk jumlah
9	Action	Text	Informasi header tabel untuk Action yang tersedia
10	View	Button	Menu untuk melihat informasi detail
11	Edit	Button	Menu untuk mengubah data kas master
12	Delete	Button	Menu untuk menghapus data kas master

5.1.5.6 Antarmuka penerimaan barang

Perancangan antarmuka list sales order akan dijelaskan pada Gambar 5.13 ,dan penejelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.11



Gambar 5.13 Perancangan antarmuka penerimaan barang

Tabel 5.11 Penjelasan antarmuka penerimaan barang

No	Nama Objek	Tipe	Keterangan
1	Judul Aplikasi	Logo button	Menampilkan judul aplikasi berupa logo
2	Nama Pengguna	Menu dropdown	Menu untuk menampilkan username dan logout user
3	Menu Utama	Menu	Menampilkan menu yang dapat diakses oleh user
4	Add	Menu	Menu untuk melakukan pembuatan data pada penerimaan barang
5	Search	Form	Melakukan pencarian data pada penerimaan barang
6	No BPU	Text	Informasi header tabel untuk No BPU
7	No PO	Text	Informasi header tabel untuk No PO
8	No ref	Text	Informasi header tabel untuk No Ref
9	Gudang	Text	Informasi header tabel untuk Gudang
10	Action	Text	Informasi header tabel untuk Action yang tersedia
11	Edit	Button	Menu untuk mengubah data penerimaan barang
12	Delete	Button	Menu untuk menghapus data penerimaan barang
13	Tambah Sepeda	Button	Menu untuk menambah item sepeda pada penerimaan barang
14	Lock	Button	Menu untuk mengunci data penerimaan barang

5.2 Implementasi Sistem

Proses implementasi dilakukan setelah selesai proses perancangan. Implementasi dilakukan berdasarkan hasil yang diperoleh dari tahap analisis kebutuhan dan tahap perancangan. Setiap kebutuhan yang telah digali dan didapatkan akan diimplementasikan pada sistem. Implementasi harus memuat struktur dan kode program berdasarkan kelas-kelas dan algoritma yang dibuat pada bagian perancangan. Pada bagian implementasi sistem ini akan menjelaskan mengenai spesifikasi sistem yang telah diimplementasikan, batasan implementasi, implementasi database, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan untuk membangun sistem ini meliputi spesifikasi perangkat keras, perangkat lunak dan sistem operasi.

5.2.1.1 Spesifikasi Perangkat Keras

Pembuatan Sistem Aplikasi Terintegrasi pada PT. Mega Abadi Motorindo menggunakan perangkat keras berupa komputer dengan spesifikasi yang dijelaskan pada Tabel 5.12.

Tabel 5.12 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
System Model	Asus A455L
Processor	Intel core i7
Memory	8,00 GB
Display	NVIDIA GeForce 940m
Hardisk	Samsung HDD
Keyboard	Logitech K240
Mouse	Logitech M212

5.2.1.2 Spesifikasi Perangkat Lunak

Pembuatan Sistem Aplikasi Terintegrasi pada PT. Mega Abadi Motorindo menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 5.13

Tabel 5.13 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Editor Perancangan	Visual Paradigm
Editor Pemrograman	Visual Studio Code, Sublime Text 3
Framework dan Bahasa Pemrograman	PHP, CSS, Javascript, Laravel
DBMS	MySQL
Aplikasi Pendukung Sistem	XAMPP
Browser	Waterfox 56.1.0 (64bit)
Database Server	phpMyAdmin

5.2.1.3 Spesifikasi Sistem Operasi

Pembuatan Sistem Aplikasi Terintegrasi pada PT. Mega Abadi Motorindo menggunakan sistem operasi dengan spesifikasi yang dijelaskan pada Tabel 5.14.

Tabel 5.14 Spesifikasi Sistem Operasi

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 8.1

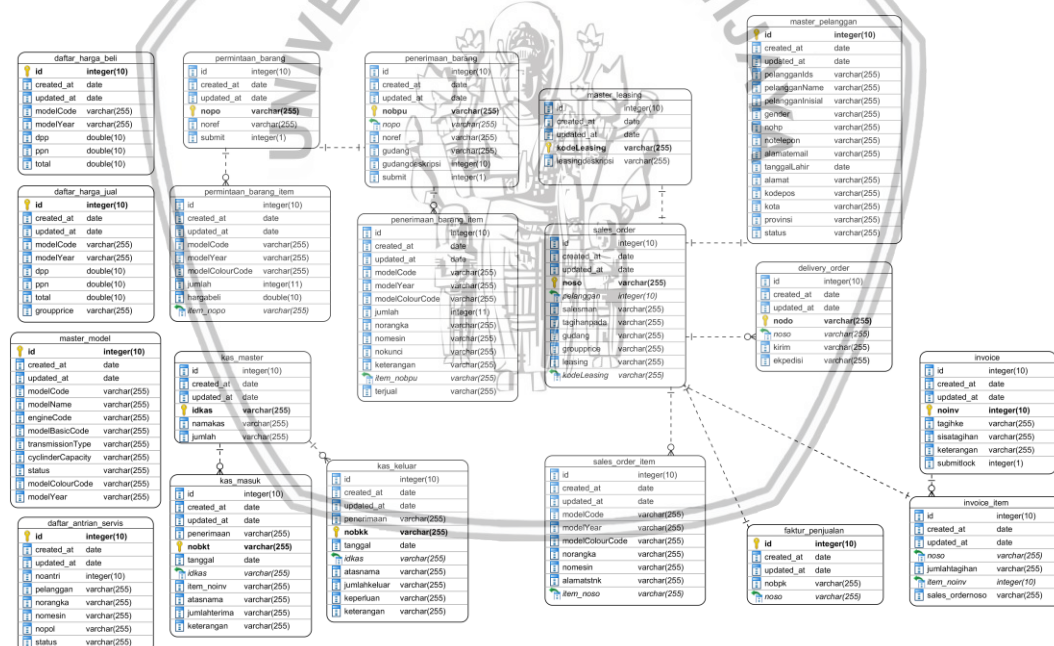
5.2.2 Batasan Implementasi

Terdapat batasan-batasan pada proses implementasi sistem ini yaitu sebagai berikut:

1. Sistem yang dibangun ini menggunakan bahasa pemrograman PHP dan javascript yang diimplementasikan pada Framework Laravel.
2. Pembuatan antarmuka sistem menggunakan HTML, CSS, dan JavaScript

5.2.3 Implementasi Database

Implementasi database dibuat berdasarkan perancangan data yang telah dibuat. Implementasi database dijelaskan pada Gambar 5.14.



Gambar 5.14 Implementasi Database

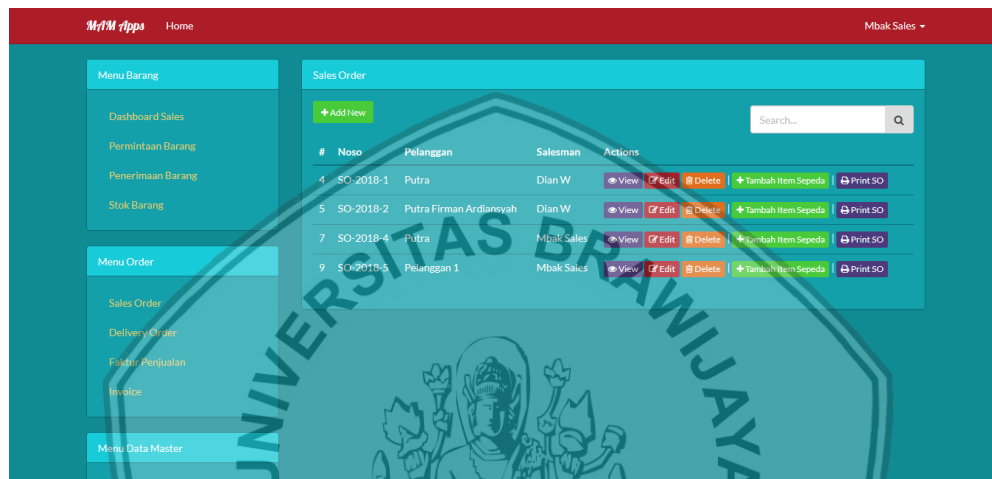
Dari implementasi data yang dijelaskan pada Gambar 5.14, diperoleh tabel - tabel dari database yang digunakan dalam membangun sistem. Pada tabel - tabel tersebut dapat dilihat bahwa ada beberapa tabel yang saling berhubungan satu sama lain, dan ada pula yang tidak. Tabel-tabel yang saling berhubungan tersebut menghasilkan *foreign key*. Contoh tabel-tabel yang saling berhubungan adalah *sales_order* yang berhubungan dengan *sales_order_item* dan menghasilkan *foreign key* "item_noso" pada tabel *sales_order_item*. Dari hubungan tersebut menjelaskan bahwa satu field pada tabel *sales_order* bisa dimiliki oleh banyak field

pada tabel sales_order_item. Beberapa tabel yang tidak berhubungan antara lain daftar_harga_beli, daftar_harga_jual, master_model, daftar_antrian_servis. Tabel yang tidak memiliki hubungan tersebut tidak memiliki *foreign key*.

5.2.4 Implementasi Antarmuka

Implementasi antarmuka ini dilakukan berdasarkan perancangan antarmuka yang dibuat pada Bab Perancangan. Berikut adalah halaman antarmuka Sistem Aplikasi Terintegrasi pada PT. Mega Abadi Motorindo.

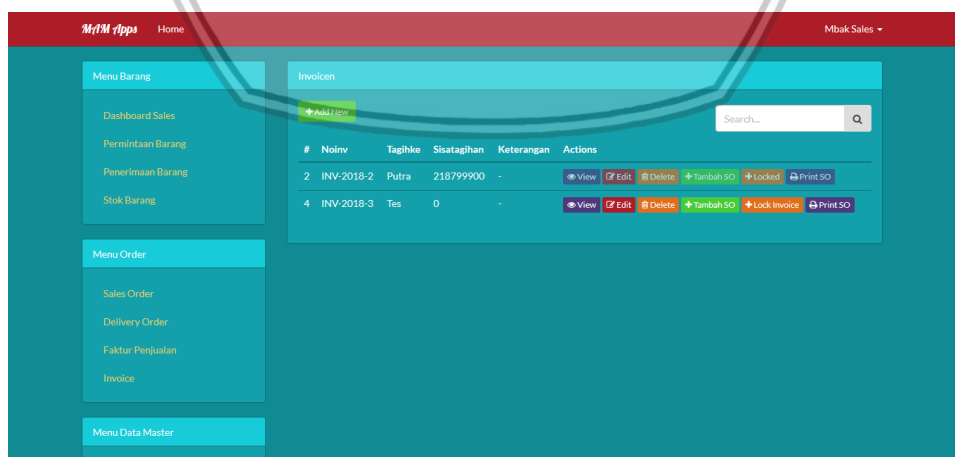
5.2.4.1 Implementasi antarmuka list sales order



Gambar 5.15 Implementasi antarmuka list sales order

Pada implementasi antarmuka list sales order seperti yang digambarkan pada Gambar 5.15 menjelaskan bahwa dalam list sales order dapat melihat detail sales order yang telah dimasukkan.

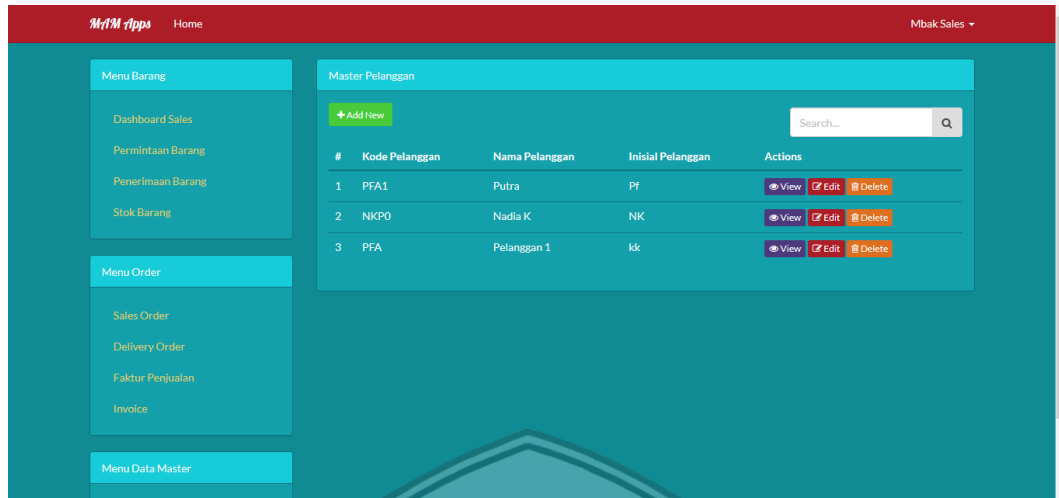
5.2.4.2 Implementasi antarmuka invoice



Gambar 5.16 Implementasi antarmuka invoice

Pada implementasi antarmuka invoice seperti yang digambarkan pada Gambar 5.16 menjelaskan bahwa dalam antarmuka invoice dapat melihat detail invoice yang telah dimasukkan.

5.2.4.3 Implementasi antarmuka master pelanggan



Gambar 5.17 Implementasi antarmuka master pelanggan

Pada implementasi antarmuka master pelanggan seperti yang digambarkan pada Gambar 5.17 menjelaskan bahwa dalam antarmuka master pelanggan dapat melihat data pelanggan yang telah dimasukkan.

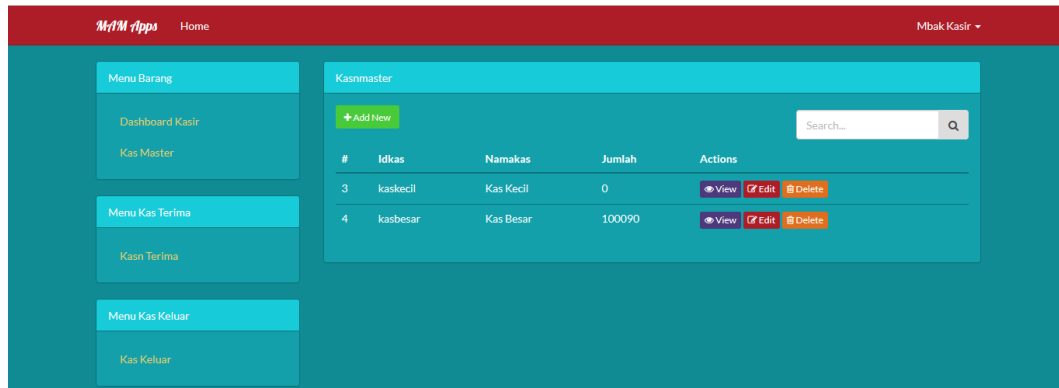
5.2.4.4 Implementasi antarmuka faktur penjualan



Gambar 5.18 Implementasi antarmuka faktur penjualan

Pada implementasi antarmuka faktur pelanggan seperti yang digambarkan pada Gambar 5.18 menjelaskan bahwa dalam antarmuka faktur pelanggan dapat melihat data faktur yang telah dimasukkan

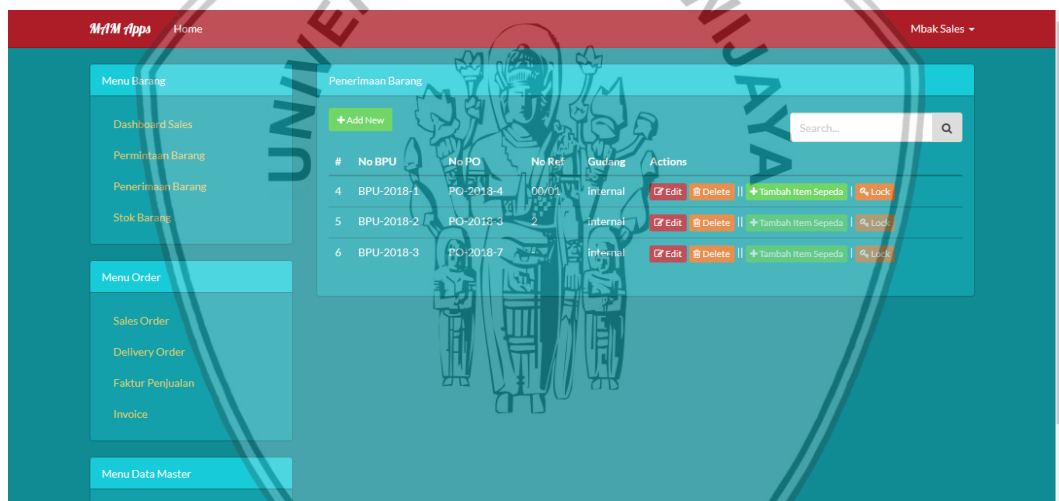
5.2.4.5 Implementasi antarmuka kas master



Gambar 5.19 Implementasi antarmuka kas master

Pada implementasi antarmuka kas master seperti yang digambarkan pada Gambar 5.19 menjelaskan bahwa dalam antarmuka kas master dapat melihat data kas yang telah dimasukan

5.2.4.6 Implementasi antarmuka penerimaan barang



Gambar 5.20 Implementasi antarmuka penerimaan barang

Pada implementasi antarmuka penerimaan barang seperti yang digambarkan pada Gambar 5.20 menjelaskan bahwa dalam antarmuka penerimaan barang dapat melihat data barang yang telah diterima.

5.2.5 Pembahasan Hasil Implementasi

Pada tahap implementasi sistem yang telah dilakukan menghasilkan spesifikasi sistem, batasan implementasi, implementasi database, implementasi antarmuka. Pada spesifikasi sistem menghasilkan spesifikasi berupa spesifikasi perangkat lunak, perangkat keras, dan sistem operasi. Spesifikasi tersebut merupakan spesifikasi yang digunakan untuk membangun sistem. Pada implementasi data diperoleh struktur database dari aplikasi yang akan dibangun. Dan pada implementasi antarmuka diperoleh tampilan web yang digunakan untuk berinteraksi antara sistem dengan pengguna sistem.

BAB 6 PENGUJIAN

Pada proses pengujian sistem ini dilakukan saat implemetasi sistem selesai dilakukan. Pengujian sistem bertujuan untuk dapat memeriksa apakah implementasi sistem telah sesuai dan benar sesuai analisis kebutuhan serta perancangan sistem. Dalam melaksanakan pengujian, dapat beberapa tahap yang harus dilakukan meliputi pengujian unit, pengujian integrasi, dan pengujian validasi.

6.1 Pengujian Unit

Pengujian ini berfokus pada unit (komponen, object, atau fungsi) dari perangkat lunak yang sedang diimplementasikan. Pada pengujian unit ini menggunakan metode white box testing dengan pengujian basis path. Pengujian unit ini akan dilakukan pada tiga sampel uji yaitu pada kelas InvoiceController, PrintController, dan AutoComplateController.

6.1.1 Pengujian Unit Klas InvoiceController Untuk Operasi LockInvoice

1. Pseudocode

```
inisialisasi variabel invoiceitem
inisialisasi variabel totaltagihan
for variabel invoiceitem as item
    set variabel totaltagihan
update tabel submitlock
update tabel sisatagihan
mengembalikan nilai return
```

1

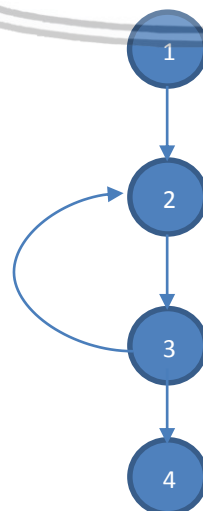
2

3

4

2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- $V(G) = 2$, ada 2 region
- $V(G) = 4 \text{ edges} - 4 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate note} + 1 = 2$

c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 2 – 4
- Jalur 2 = 1 – 2 – 4

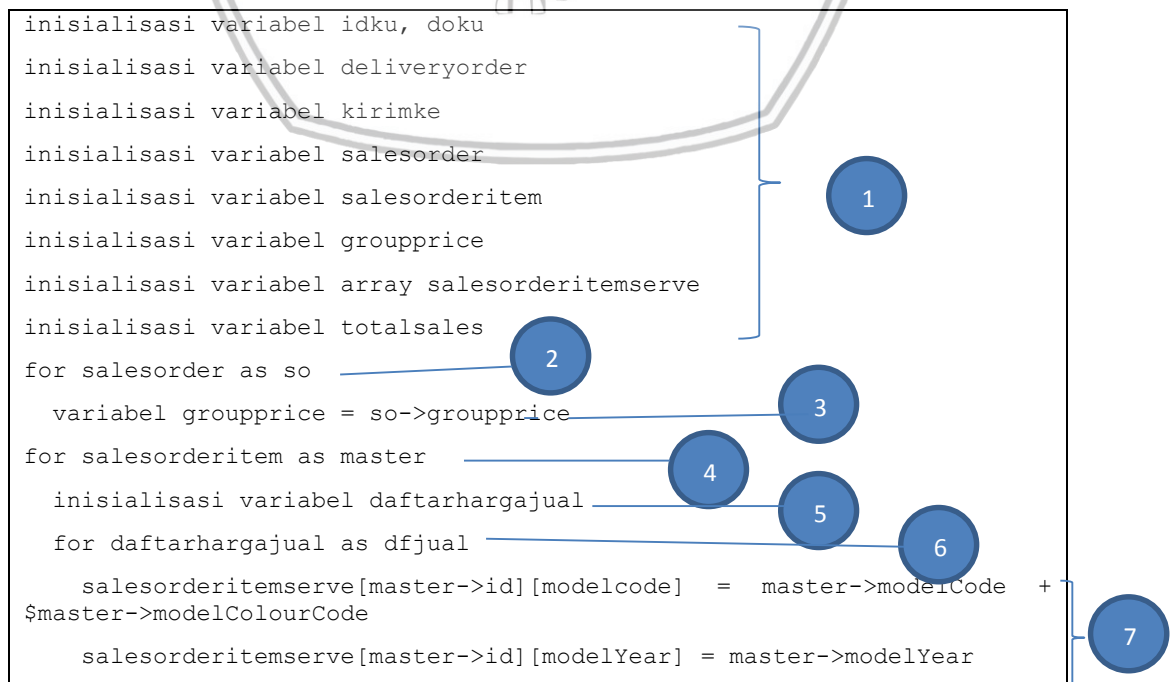
Test case dan hasil akan dijelaskan pada Tabel 6.1

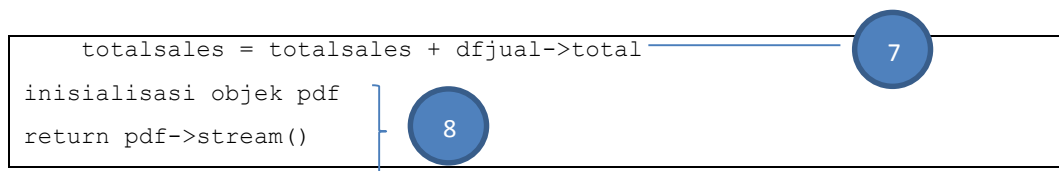
Tabel 6.1 Hasil Pengujian Unit Kelas InvoiceController Untuk Operasi LockInvoice

No	No. Jalur	Prosedur Uji	Expected result	Result	Status
1	1	Memanggil operasi LockInvoice dengan item_noinv = INV-2018-3	Berhasil mengganti status invoice menjadi lock	Berhasil mengganti status invoice menjadi lock	Valid
2	2	Memanggil operasi LockInvoice dengan item_noinv = DO-2018-3	Gagal, status invoice tidak berubah	Gagal, status invoice tidak berubah	Valid

6.1.2 Pengujian Unit Klas PrintController Untuk Operasi PrintDeliveryOrder

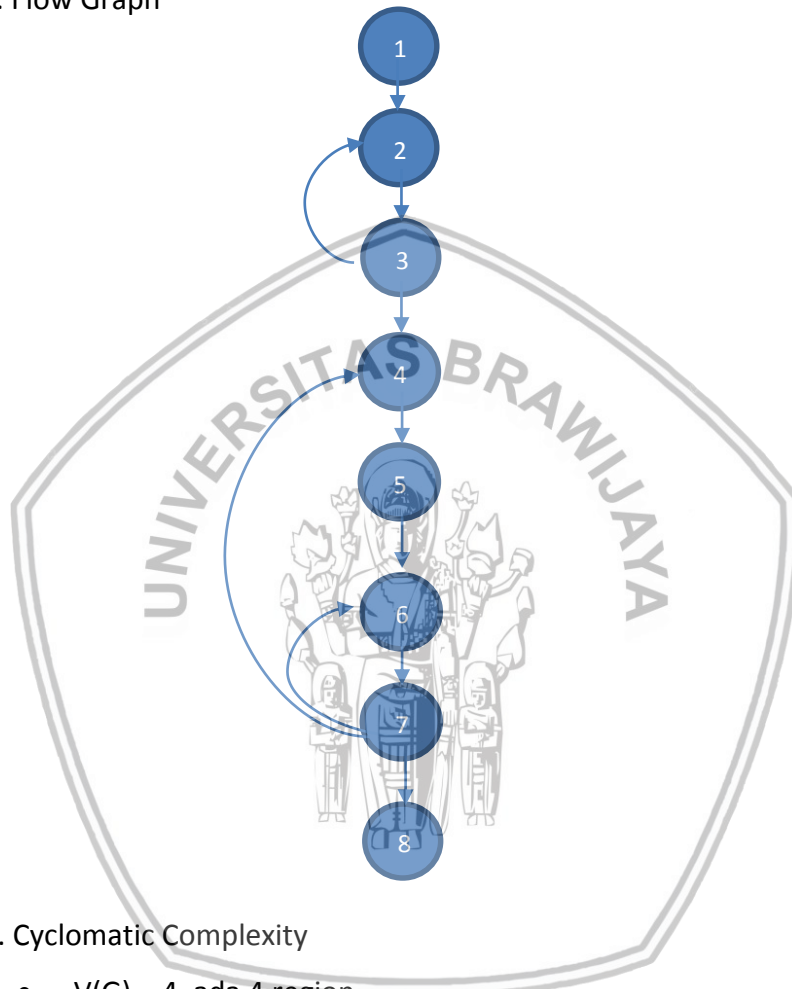
1. Pseudocode





2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- $V(G) = 4$, ada 4 region
- $V(G) = 10 \text{ edges} - 8 \text{ nodes} + 2 = 4$
- $V(G) = 3 \text{ predicate note} + 1 = 4$

c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 2 – 4 – 5 – 6 – 7 – 4 – 8
- Jalur 2 = 1 – 2 – 4 – 5 – 6 – 7 – 4 – 8
- Jalur 3 = 1 – 2 – 3 – 2 – 4 – 8
- Jalur 4 = 1 – 2 – 4 – 8

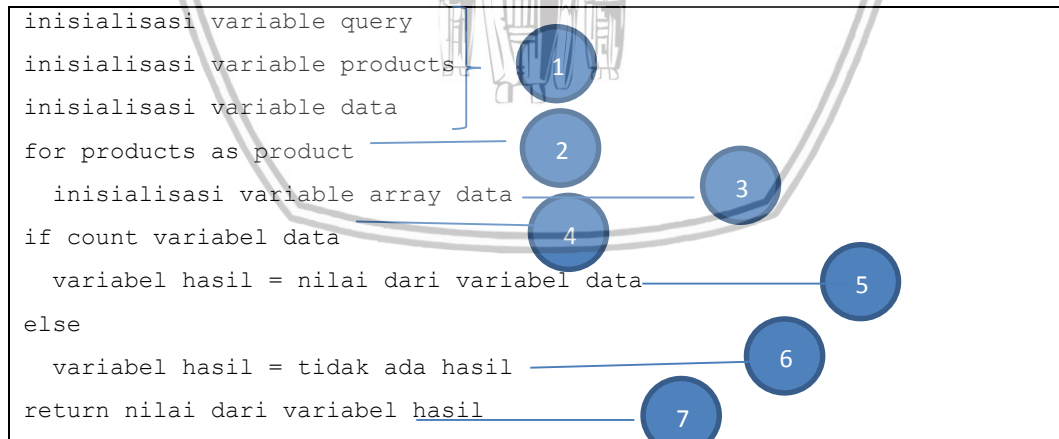
Test case dan hasil akan dijelaskan pada Tabel 6.2

Tabel 6.2 Hasil Pengujian Unit Kelas PrintController Untuk Operasi PrintDeliveryOrder

No	No. Jalur	Prosedur Uji	Expected result	Result	Status
1	1	Memanggil operasi PrintDeliveryOrder dengan parameter id_salesorder = 0 dan id_deliveryorder = 0	Mengembalikan hasil berupa data pdf yang berisi detail deliveryorder	Mengembalikan hasil berupa data pdf yang berisi detail deliveryorder	Valid
2	2	Memanggil operasi PrintDeliveryOrder dengan parameter id_salesorder = x dan id_deliveryorder = 0	Mengembalikan data pdf dengan tidak ada detail	Mengembalikan data pdf dengan tidak ada detail	Valid

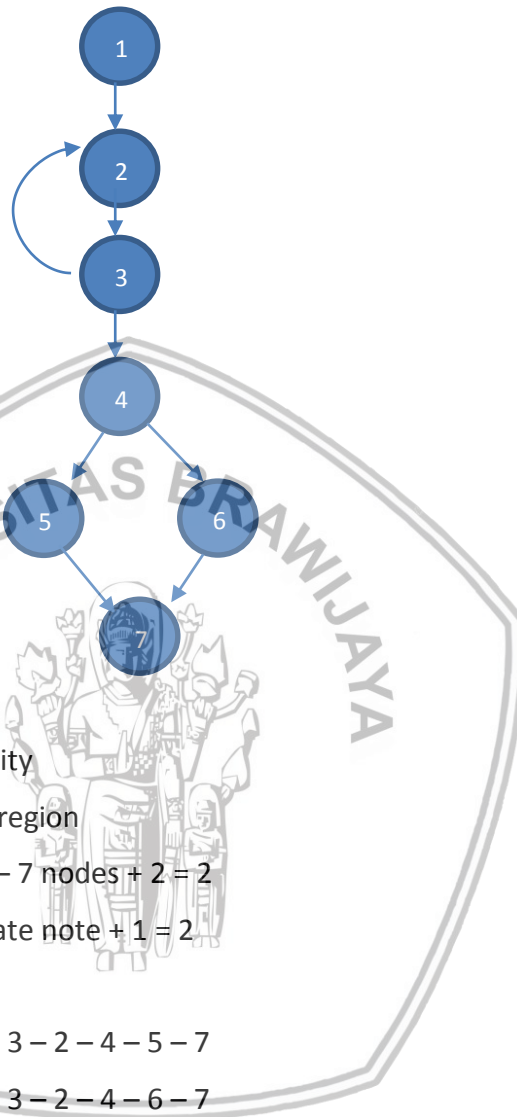
6.1.3 Pengujian Unit Klas AutoComplateController Untuk Operasi SearchNoInvoiceWithTagihan

1. Pseudocode



2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- $V(G) = 2$, ada 2 region
- $V(G) = 7 \text{ edges} - 7 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate node} + 1 = 2$

c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 2 – 4 – 5 – 7
- Jalur 2 = 1 – 2 – 3 – 2 – 4 – 6 – 7
- Jalur 3 = 1 – 2 – 3 – 4 – 6

Test case dan hasil akan dijelaskan pada Tabel 6.3

Tabel 6.3 Hasil Pengujian Unit Kelas autoCompleteController Untuk Operasi SearchNoInvoiceWithTagihan

No	No. Jalur	Prosedur Uji	Expected result	Result	Status
1	1	Memanggil operasi SearchNoInvoiceWithTagihan dengan noinv = INV	Mengembalikan hasil berupa array yang berisi detail tagihan invoice	Mengembalikan hasil berupa array yang berisi detail tagihan invoice	Valid

2	2	Memanggil operasi SearchNoInvoiceWithTagihan dengan noinv = INV-2018-4-	Nilai kembalian berupa “tidak ada hasil”	Nilai kembalian berupa “tidak ada hasil”	Valid
3	3	Memanggil operasi SearchNoInvoiceWithTagihan dengan noinv = SO-2018-3	Nilai kembalian berupa “tidak ada hasil”	Nilai kembalian berupa “tidak ada hasil”	Valid

6.2 Pengujian Integrasi

Dalam pengujian integrasi, setiap klas yang memiliki hubungan satu dengan yang lainnya dilakukan pengujian. Dalam pengujian ini, basis path testing digunakan sebagai teknik pengujian. Hanya akan terdapat tiga operasi pada suatu klas yang menggunakan operasi-operasi pada klas lain yang akan dilakukan pengujian untuk dijadikan sampel uji.

Operasi pertama adalah operasi sendPermintaanBarangKePusat() pada klas EmailController yang memanggil operasi send() dan failure() pada klas Mail

Operasi pertama adalah operasi searchNoPo() pada klas AutoComplateController yang memanggil operasi get() pada klas Request

Operasi pertama adalah operasi cronTaskUltah() pada klas CrmDashboardController yang memanggil operasi api_wa_send_plain() pada klas CrmDashboardController

6.2.1 Pengujian Integrasi Klas EmailController Untuk Operasi sendPermintaanBarangKePusat

1. Pseudocode

```
inisialisasi variabel permintaanbarang = melakukan query database pada
tabel permintaanbarang
inisialisasi variabel permintaanbarangitem = melakukan query database
pada tabel permintaanbarangitem
melakukan query database pada tabel permintaanbarang untuk merubah status
permintaan barang menjadi telah terkirim
memanggil fungsi send() pada class Mail dengan menggunakan parameter
variabel permintaanbarang, variabel permintaanbarangitem dan email
tujuan.
If (count(Mail::failures()) > 0)
    Inisialisasi variable error = Failed to send password reset email,
    please try again.
Else
    Inisialisasi variable error = email send
mengembalikan variabel error untuk melakukan redirect ke halaman
Permintaan Barang
```

1

2

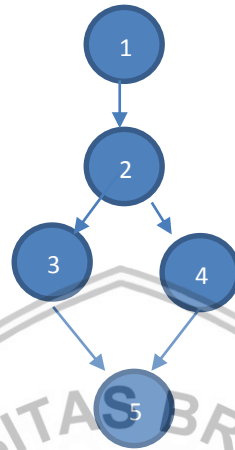
3

4

5

2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- $V(G) = 2$, ada 2 region
- $V(G) = 5 \text{ edges} - 5 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate node} + 1 = 2$

c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 5
- Jalur 2 = 1 – 2 – 4 – 5

Test case dan hasil akan dijelaskan pada Tabel 6.4

Tabel 6.4 Hasil Pengujian Integrasi Kelas Controller Untuk Operasi

No	No. Jalur	Prosedur Uji	Expected result	Result	Status
1	1	Memanggil operasi sendPermintaanBarangKePusat dengan dengan parameter = PO-2018-5	Melakukan pengiriman data permintaan ke pusat	Melakukan pengiriman data permintaan ke pusat	Valid
2	2	Memanggil operasi sendPermintaanBarangKePusat dengan dengan parameter = SO-2018-5	Tidak melakukan pengiriman data ke pusat	Tidak melakukan pengiriman data ke pusat	Valid

6.2.2 Pengujian Integrasi Klas AutoComplateController Untuk Operasi searchNoPo

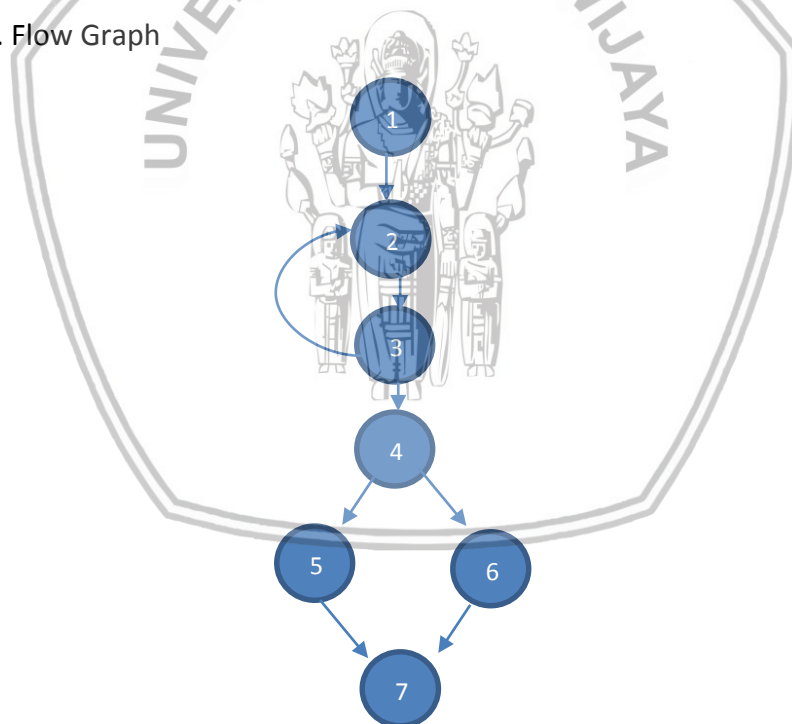
1. Pseudocode

```

Inisialisasi variabel query = memanggil fungsi get() pada request dengan parameter term
Melakukan query ke database pada tabel permintaan_barang
Inisialisasi array data
Foreach(products as product) _____ 2
    Array data = array(value=>product->no, id=>product->id) _____ 3
If(count(data)) _____ 4
    Variabel hasil = variabel data _____ 5
Else _____ 6
    Variabel hasil = nilai no result found _____
Mengembalikan nilai hasil _____ 7
    
```

2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- $V(G) = 2$, ada 2 region
- $V(G) = 7 \text{ edges} - 7 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate note} + 1 = 2$

c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 2 – 4 – 5 – 7
- Jalur 2 = 1 – 2 – 3 – 2 – 4 – 6 – 7
- Jalur 3 = 1 – 2 – 3 – 4 – 6 – 7

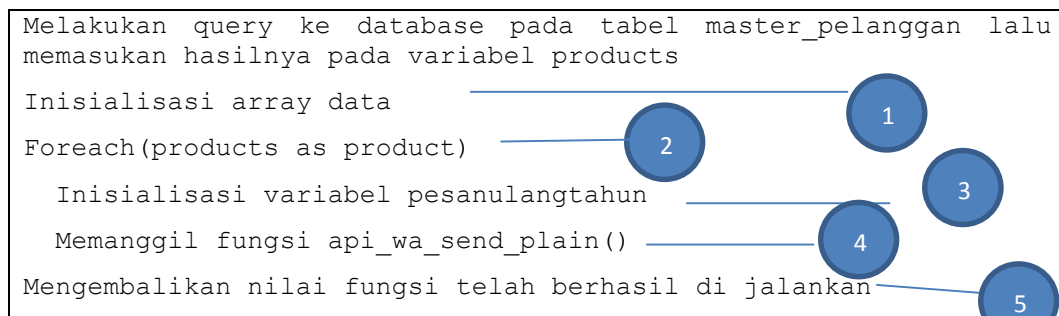
Test case dan hasil akan dijelaskan pada Tabel 6.5

Tabel 6.5 Hasil Pengujian Integrasi Kelas Controller Untuk Operasi

No	No. Jalur	Prosedur Uji	Expected result	Result	Status
1	1	Memanggil operasi searchNoPo dengan dengan variabel request->term = PO-2018-3	Mengembalikan hasil berupa array yang berisi detail permintaan barang	Mengembalikan hasil berupa array yang berisi detail permintaan barang	Valid
2	2	Memanggil operasi searchNoPo dengan dengan variabel request->term = PO-2018-300	Nilai kembalian berupa “tidak ada hasil”	Nilai kembalian berupa “tidak ada hasil”	Valid
3	3	Memanggil operasi searchNoPo dengan dengan variabel request->term = IBV-2018-3	Nilai kembalian berupa “tidak ada hasil”	Nilai kembalian berupa “tidak ada hasil”	Valid

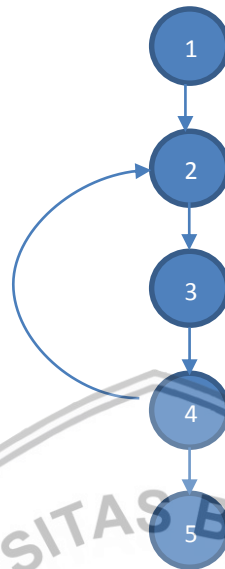
6.2.3 Pengujian Integrasi Klas CrmDashboardController Untuk Operasi cronTaskUltah

1. Pseudocode



2. Basis Path Testing

a. Flow Graph



b. Cyclomatic Complexity

- $V(G) = 2$, ada 2 region
- $V(G) = 5 \text{ edges} - 5 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate node} + 1 = 2$

c. Independent Path

- Jalur 1 = 1 – 2 – 3 – 4 – 2 – 5
- Jalur 2 = 1 – 2 – 5

Test case dan hasil akan dijelaskan pada Tabel 6.6

Tabel 6.6 Hasil Pengujian Integrasi Kelas Controller Untuk Operasi

No	No. Jalur	Prosedur Uji	Expected result	Result	Status
1	1	Memanggil operasi cronTaskUltah pada tanggal ulang tahun pelanggan	Mengimkan pesan ulang tahun kepada pelanggan	Mengimkan pesan ulang tahun kepada pelanggan	Valid
2	2	Memanggil operasi cronTaskUltah tidak pada tanggal ulang tahun pelanggan	Tidak mengirimkan pesan	Tidak mengirimkan pesan	Valid

6.3 Pengujian Validasi

Pada tahap ini akan dilakukan pengujian validasi yang bertujuan untuk mengetahui dan menyatakan bahwa sistem yang dibuat telah sesuai dengan yang diharapkan. Pengujian validasi dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik dan tidak ada kesalahan yang terjadi. Pada pengujian validasi ini digunakan metode black box testing. Dengan black box testing memungkinkan pembuat perangkat lunak untuk memperoleh kondisi yang terjadi untuk suatu masukan yang akan menjalankan semua kebutuhan.

6.3.1 Pengujian Validasi Login

- a. Kasus uji berhasil login sebagai sales

Tabel 6.7 Kasus uji berhasil login sebagai sales

Nama Kasus Uji	Kasus Uji berhasil login sebagai sales
Prosedur	1. Membuka halaman login 2. Memasukan email = sales@mail.com dan password = salessales 3. Menekan tombol sign in
Hasil yang diharapkan	Sistem menampilkan halaman Dashboard Sales
Hasil	Sistem menampilkan halaman Dashboard Sales
Status	Valid

- b. Kasus uji berhasil login sebagai manager

Tabel 6.8 Kasus uji berhasil login sebagai manager

Nama Kasus Uji	Kasus Uji berhasil login sebagai manger
Prosedur	1. Membuka halaman login 2. Memasukan email = manager@mail.com dan password = manager 3. Menekan tombol sign in
Hasil yang diharapkan	Sistem menampilkan halaman Dashboard Manager
Hasil	Sistem menampilkan halaman Dashboard Manager
Status	Valid

c. Kasus uji berhasil login sebagai admin

Tabel 6.9 Kasus uji berhasil login sebagai admin

Nama Kasus Uji	Kasus Uji berhasil login sebagai admin
Prosedur	1. Membuka halaman login 2. Memasukan email = admin@mail.com dan password = superadmin 3. Menekan tombol sign in
Hasil yang diharapkan	Sistem menampilkan halaman Dashboard Admin
Hasil	Sistem menampilkan halaman Dashboard Admin
Status	Valid

d. Kasus uji gagal login

Tabel 6.10 Kasus uji gagal login

Nama Kasus Uji	Kasus Uji gagal login
Prosedur	1. Membuka halaman login 2. Memasukan email = me@mail.com dan password = superadmin 3. Menekan tombol sign in
Hasil yang diharapkan	Sistem tetap menampilkan halaman login
Hasil	Sistem tetap menampilkan halaman login
Status	Valid

6.3.2 Pengujian Validasi Logout

a. Kasus uji logout

Tabel 6.11 Kasus uji logout

Nama Kasus Uji	Kasus Uji logout
Prosedur	1. Sudah berada dalam kondisi logon 2. Menekan tombol logout pada menu dropdown profile
Hasil yang diharapkan	Keluar dari kondisi logon dan menampilkan halaman login
Hasil	Keluar dari kondisi logon dan menampilkan halaman login
Status	Valid

6.3.3 Pengujian Validasi Melihat master data motor

- a. Kasus uji melihat master data motor

Tabel 6.12 Kasus uji Melihat master data motor

Nama Kasus Uji	Kasus Uji melihat master data motor
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman master data
Hasil yang diharapkan	Menampilkan master data motor
Hasil	Menampilkan master data motor
Status	Valid

6.3.4 Pengujian Validasi Membuat master data motor

- a. Kasus uji membuat master data motor

Tabel 6.13 Kasus uji membuat master data motor

Nama Kasus Uji	Kasus Uji membuat master data motor
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman master data 3. Menekan tombol add 4. Memasukkan form pada master data 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada master data motor
Hasil	Berhasil menyimpan dan memasukkan data pada master data motor
Status	Valid

6.3.5 Pengujian Validasi Mengubah master data motor

- a. Kasus uji mengubah master data motor

Tabel 6.14 Kasus uji mengubah master data motor

Nama Kasus Uji	Kasus Uji mengubah master data motor
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman master data 3. Menekan tombol edit

	4. Memasukkan data yang baru pada form master data 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada master data motor
Hasil	Berhasil menyimpan dan mengubah data pada master data motor
Status	Valid

6.3.6 Pengujian Validasi Menghapus master data motor

- a. Kasus uji menghapus master data motor

Tabel 6.15 Kasus uji menghapus master data motor

Nama Kasus Uji	Kasus Uji menghapus master data motor
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman master data 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.7 Pengujian Validasi Melihat daftar harga beli

- a. Kasus uji melihat daftar harga beli

Tabel 6.16 Kasus uji Melihat daftar harga beli

Nama Kasus Uji	Kasus Uji melihat daftar harga beli
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga beli
Hasil yang diharapkan	Menampilkan daftar harga beli
Hasil	Menampilkan daftar harga beli
Status	Valid

6.3.8 Pengujian Validasi Menambah daftar harga beli

a. Kasus uji Menambah daftar harga beli

Tabel 6.17 Kasus uji Menambah daftar harga beli

Nama Kasus Uji	Kasus Uji Menambah daftar harga beli
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga beli 3. Menekan tombol add 4. Memasukkan form pada daftar harga beli 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada daftar harga beli
Hasil	Berhasil menyimpan dan memasukkan data pada daftar harga beli
Status	Valid

6.3.9 Pengujian Validasi Mengubah daftar harga beli

a. Kasus uji mengubah daftar harga beli

Tabel 6.18 Kasus uji mengubah daftar harga beli

Nama Kasus Uji	Kasus Uji mengubah daftar harga beli
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga beli 3. Menekan tombol edit 4. Memasukkan data yang baru pada form daftar harga beli 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada daftar harga beli
Hasil	Berhasil menyimpan dan mengubah data pada daftar harga beli
Status	Valid

6.3.10 Pengujian Validasi Menghapus daftar harga beli

- a. Kasus uji menghapus daftar harga beli

Tabel 6.19 Kasus uji menghapus daftar harga beli

Nama Kasus Uji	Kasus Uji menghapus daftar harga beli
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga beli 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.11 Pengujian Validasi Melihat daftar harga jual

- a. Kasus uji melihat daftar harga jual

Tabel 6.20 Kasus uji Melihat daftar harga jual

Nama Kasus Uji	Kasus Uji melihat daftar harga jual
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga jual
Hasil yang diharapkan	Menampilkan daftar harga jual
Hasil	Menampilkan daftar harga jual
Status	Valid

6.3.12 Pengujian Validasi Menambah daftar harga jual

- a. Kasus uji Menambah daftar harga jual

Tabel 6.21 Kasus uji Menambah daftar harga jual

Nama Kasus Uji	Kasus Uji Menambah daftar harga jual
Prosedur	1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga jual 3. Menekan tombol add 4. Memasukkan form pada daftar harga jual 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada daftar harga jual

Hasil	Berhasil menyimpan dan memasukkan data pada daftar harga jual
Status	Valid

6.3.13 Pengujian Validasi Mengubah daftar harga jual

- a. Kasus uji mengubah daftar harga jual

Tabel 6.22 Kasus uji mengubah daftar harga jual

Nama Kasus Uji	Kasus Uji mengubah daftar harga jual
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga jual 3. Menekan tombol edit 4. Memasukkan data baru pada form daftar harga jual 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada daftar harga jual
Hasil	Berhasil menyimpan dan memasukkan data pada daftar harga jual
Status	Valid

6.3.14 Pengujian Validasi Menghapus daftar harga jual

- a. Kasus uji menghapus daftar harga jual

Tabel 6.23 Kasus uji menghapus daftar harga jual

Nama Kasus Uji	Kasus Uji menghapus daftar harga jual
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai admin 2. Masuk ke dalam halaman daftar harga jual 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.15 Pengujian Validasi Melihat daftar permintaan barang

- a. Kasus uji melihat daftar permintaan barang

Tabel 6.24 Kasus uji Melihat daftar permintaan barang

Nama Kasus Uji	Kasus Uji melihat daftar permintaan barang
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar permintaan barang
Hasil yang diharapkan	Menampilkan daftar permintaan barang
Hasil	Menampilkan daftar permintaan barang
Status	Valid

6.3.16 Pengujian Validasi Membuat permintaan barang baru

a. Kasus uji membuat permintaan barang baru

Tabel 6.25 Kasus uji membuat permintaan barang baru

Nama Kasus Uji	Kasus Uji membuat permintaan barang baru
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar permintaan barang baru 3. Menekan tombol add 4. Memasukkan form pada permintaan barang baru 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada permintaan barang baru
Hasil	Berhasil menyimpan dan memasukkan data pada permintaan barang baru
Status	Valid

6.3.17 Pengujian Validasi Mengubah permintaan barang baru

a. Kasus uji mengubah permintaan barang baru

Tabel 6.26 Kasus uji mengubah permintaan barang baru

Nama Kasus Uji	Kasus Uji mengubah permintaan barang baru
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar permintaan barang baru 3. Menekan tombol edit

	4. Memasukkan data yang baru pada form permintaan barang baru 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada permintaan barang baru
Hasil	Berhasil menyimpan dan mengubah data pada permintaan barang baru
Status	Valid

6.3.18 Pengujian Validasi Menghapus permintaan barang baru

- a. Kasus uji menghapus permintaan barang baru

Tabel 6.27 Kasus uji menghapus permintaan barang baru

Nama Kasus Uji	Kasus Uji menghapus permintaan barang baru
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman permintaan barang baru 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.19 -Pengujian Validasi Mengirim informasi permintaan barang baru

- a. Kasus uji -

Tabel 6.28 Kasus uji menghapus permintaan barang baru

Nama Kasus Uji	Kasus Uji menghapus permintaan barang baru
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman permintaan barang baru 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.20 Pengujian Validasi Membuat bukti penerimaan barang

a. Kasus uji Membuat bukti penerimaan barang

Tabel 6.29 Kasus uji Membuat bukti penerimaan barang

Nama Kasus Uji	Kasus Uji Membuat bukti penerimaan barang
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar penerimaan barang baru 3. Menekan tombol add 4. Memasukkan form pada penerimaan barang baru 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada penerimaan barang baru
Hasil	Berhasil menyimpan dan memasukkan data pada penerimaan barang baru
Status	Valid

6.3.21 Pengujian Validasi Melihat bukti penerimaan barang

a. Kasus uji melihat bukti penerimaan barang

Tabel 6.30 Kasus uji Melihat bukti penerimaan barang

Nama Kasus Uji	Kasus Uji melihat daftar bukti penerimaan barang
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman bukti penerimaan barang
Hasil yang diharapkan	Menampilkan bukti penerimaan barang
Hasil	Menampilkan bukti penerimaan barang
Status	Valid

6.3.22 Pengujian Validasi Mengubah bukti penerimaan barang

a. Kasus uji mengubah penerimaan barang baru

Tabel 6.31 Kasus uji mengubah penerimaan barang baru

Nama Kasus Uji	Kasus Uji mengubah penerimaan barang baru
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales

	2. Masuk ke dalam halaman daftar penerimaan barang baru 3. Menekan tombol edit 4. Memasukkan data yang baru pada form penerimaan barang baru 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada penerimaan barang baru
Hasil	Berhasil menyimpan dan mengubah data pada penerimaan barang baru
Status	Valid

6.3.23 Pengujian Validasi Menghapus bukti penerimaan barang

- a. Kasus uji menghapus bukti penerimaan barang baru

Tabel 6.32 Kasus uji menghapus bukti penerimaan barang baru

Nama Kasus Uji	Kasus Uji menghapus bukti penerimaan barang baru
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman bukti penerimaan barang baru 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.24 Pengujian Validasi Mengunci status bukti penerimaan barang

- a. Kasus uji Mengunci status bukti penerimaan barang

Tabel 6.33 Kasus uji Mengunci status bukti penerimaan barang

Nama Kasus Uji	Kasus Uji Mengunci status bukti penerimaan barang
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman bukti penerimaan barang baru 3. Menekan tombol lock
Hasil yang diharapkan	Data berhasil dikunci
Hasil	Data berhasil dikunci

Status	Valid
--------	-------

6.3.25 Pengujian Validasi Melihat jumlah stok barang

- a. Kasus uji Melihat jumlah stok barang

Tabel 6.34 Kasus uji Melihat jumlah stok barang

Nama Kasus Uji	Kasus Uji Melihat jumlah stok barang
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman stok barang
Hasil yang diharapkan	Menampilkan jumlah stok barang
Hasil	Menampilkan jumlah stok barang
Status	Valid

6.3.26 Pengujian Validasi Membuat order penjualan baru

- a. Kasus uji Membuat order penjualan baru

Tabel 6.35 Kasus uji Membuat order penjualan baru

Nama Kasus Uji	Kasus Uji Membuat order penjualan baru
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar order penjualan baru 3. Menekan tombol add 4. Memasukkan form pada order penjualan baru 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada order penjualan baru
Hasil	Berhasil menyimpan dan memasukkan data pada order penjualan baru
Status	Valid

6.3.27 Pengujian Validasi Melihat order penjualan

- a. Kasus uji melihat order penjualan

Tabel 6.36 Kasus uji Melihat order penjualan

Nama Kasus Uji	Kasus Uji melihat daftar order penjualan
----------------	------------------------------------------

Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman order penjualan
Hasil yang diharapkan	Menampilkan order penjualan
Hasil	Menampilkan order penjualan
Status	Valid

6.3.28 Pengujian Validasi Mengubah order penjualan

a. Kasus uji mengubah order penjualan

Tabel 6.37 Kasus uji mengubah order penjualan

Nama Kasus Uji	Kasus Uji mengubah order penjualan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar order penjualan 3. Menekan tombol edit 4. Memasukkan data yang baru pada form order penjualan 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada order penjualan
Hasil	Berhasil menyimpan dan mengubah data pada order penjualan
Status	Valid

6.3.29 Pengujian Validasi Menghapus order penjualan

a. Kasus uji menghapus order penjualan

Tabel 6.38 Kasus uji menghapus order penjualan

Nama Kasus Uji	Kasus Uji menghapus order penjualan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman order penjualan 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.30 Pengujian Validasi Mencetak detail order penjualan

- a. Kasus uji Mencetak detail order penjualan

Tabel 6.39 Kasus uji Mencetak order penjualan

Nama Kasus Uji	Kasus Uji Mencetak detail order penjualan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman order penjualan 3. Menekan tombol print
Hasil yang diharapkan	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Hasil	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Status	Valid

6.3.31 Pengujian Validasi Membuat surat jalan

- a. Kasus uji Membuat surat jalan

Tabel 6.40 Kasus uji Membuat surat jalan

Nama Kasus Uji	Kasus Uji Membuat surat jalan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar surat jalan 3. Menekan tombol add 4. Memasukkan form pada surat jalan 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada surat jalan
Hasil	Berhasil menyimpan dan memasukkan data pada surat jalan
Status	Valid

6.3.32 Pengujian Validasi Mengubah surat jalan

- a. Kasus uji mengubah surat jalan

Tabel 6.41 Kasus uji mengubah surat jalan

Nama Kasus Uji	Kasus Uji mengubah surat jalan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales

	2. Masuk ke dalam halaman daftar surat jalan 3. Menekan tombol edit 4. Memasukkan data yang baru pada form surat jalan 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada surat jalan
Hasil	Berhasil menyimpan dan mengubah data pada surat jalan
Status	Valid

6.3.33 Pengujian Validasi Melihat surat jalan

- a. Kasus uji melihat surat jalan

Tabel 6.42 Kasus uji Melihat surat jalan

Nama Kasus Uji	Kasus Uji melihat daftar surat jalan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat jalan
Hasil yang diharapkan	Menampilkan daftar surat jalan
Hasil	Menampilkan daftar surat jalan
Status	Valid

6.3.34 Pengujian Validasi Menghapus surat jalan

- a. Kasus uji menghapus surat jalan

Tabel 6.43 Kasus uji menghapus surat jalan

Nama Kasus Uji	Kasus Uji menghapus surat jalan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat jalan 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.35 Pengujian Validasi Mencetak surat jalan

- a. Kasus uji Mencetak surat jalan

Tabel 6.44 Kasus uji Mencetak surat jalan

Nama Kasus Uji	Kasus Uji Mencetak detail surat jalan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat jalan 3. Menekan tombol print
Hasil yang diharapkan	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Hasil	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Status	Valid

6.3.36 Pengujian Validasi Membuat surat faktur pajak

a. Kasus uji Membuat surat faktur pajak

Tabel 6.45 Kasus uji Membuat surat faktur pajak

Nama Kasus Uji	Kasus Uji Membuat surat faktur pajak
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar surat faktur pajak 3. Menekan tombol add 4. Memasukkan form pada surat faktur pajak 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada surat faktur pajak
Hasil	Berhasil menyimpan dan memasukkan data pada surat faktur pajak
Status	Valid

6.3.37 Pengujian Validasi Mengubah surat faktur pajak

a. Kasus uji mengubah surat faktur pajak

Tabel 6.46 Kasus uji mengubah surat faktur pajak

Nama Kasus Uji	Kasus Uji mengubah surat faktur pajak
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar surat faktur pajak

	3. Menekan tombol edit 4. Memasukkan data yang baru pada form surat faktur pajak 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada surat faktur pajak
Hasil	Berhasil menyimpan dan mengubah data pada surat faktur pajak
Status	Valid

6.3.38 Pengujian Validasi Melihat surat faktur pajak

- a. Kasus uji melihat surat faktur pajak

Tabel 6.47 Kasus uji Melihat surat faktur pajak

Nama Kasus Uji	Kasus Uji melihat daftar surat faktur pajak
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat faktur pajak
Hasil yang diharapkan	Menampilkan daftar surat faktur pajak
Hasil	Menampilkan daftar surat faktur pajak
Status	Valid

6.3.39 Pengujian Validasi Menghapus surat faktur pajak

- a. Kasus uji menghapus surat faktur pajak

Tabel 6.48 Kasus uji menghapus surat faktur pajak

Nama Kasus Uji	Kasus Uji menghapus surat faktur pajak
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat faktur pajak 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.40 Pengujian Validasi Mencetak surat faktur pajak

- a. Kasus uji Mencetak surat faktur pajak

Tabel 6.49 Kasus uji Mencetak surat faktur pajak

Nama Kasus Uji	Kasus Uji Mencetak detail surat faktur pajak
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat faktur pajak 3. Menekan tombol print
Hasil yang diharapkan	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Hasil	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Status	Valid

6.3.41 Pengujian Validasi Membuat invoice

a. Kasus uji Membuat invoice

Tabel 6.50 Kasus uji Membuat invoice

Nama Kasus Uji	Kasus Uji Membuat invoice
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar invoice 3. Menekan tombol add 4. Memasukkan form pada invoice 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada invoice
Hasil	Berhasil menyimpan dan memasukkan data pada invoice
Status	Valid

6.3.42 Pengujian Validasi Mengubah invoice

a. Kasus uji mengubah invoice

Tabel 6.51 Kasus uji mengubah invoice

Nama Kasus Uji	Kasus Uji mengubah invoice
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar invoice 3. Menekan tombol edit

	4. Memasukkan data yang baru pada form invoice 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada invoice
Hasil	Berhasil menyimpan dan mengubah data pada invoice
Status	Valid

6.3.43 Pengujian Validasi Melihat invoice

- a. Kasus uji melihat invoice

Tabel 6.52 Kasus uji Melihat invoice

Nama Kasus Uji	Kasus Uji melihat invoice
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman invoice
Hasil yang diharapkan	Menampilkan daftar invoice
Hasil	Menampilkan daftar invoice
Status	Valid

6.3.44 Pengujian Validasi Menghapus invoice

- a. Kasus uji menghapus invoice

Tabel 6.53 Kasus uji menghapus invoice

Nama Kasus Uji	Kasus Uji menghapus invoice
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman invoice 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.45 Pengujian Validasi Menambah sales order pada invoice

- a. Kasus uji Menambah sales order pada invoice

Tabel 6.54 Kasus uji Menambah sales order pada invoice

Nama Kasus Uji	Kasus Uji Menambah sales order pada invoice
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman invoice 3. Menekan tombol tambah SO 4. Sistem akan menampilkan daftar sales order 5. Menekan tombol add 6. Mengisi nomer SO yang akan di inputkan 7. Menekan tombol save
Hasil yang diharapkan	Data sales order akan tersimpan pada invoice
Hasil	Data sales order akan tersimpan pada invoice
Status	Valid

6.3.46 Pengujian Validasi Mencetak detail invoice

a. Kasus uji Mencetak detail invoice

Tabel 6.55 Kasus uji Mencetak detail invoice

Nama Kasus Uji	Kasus Uji Mencetak detail invoice
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman surat invoice 3. Menekan tombol print
Hasil yang diharapkan	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Hasil	Akan tampil data yang akan siap tercetak dalam bentuk pdf
Status	Valid

6.3.47 Pengujian Validasi Membuat master data pelanggan

a. Kasus uji Membuat master data pelanggan

Tabel 6.56 Kasus uji Membuat master data pelanggan

Nama Kasus Uji	Kasus Uji Membuat master data pelanggan
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar master data pelanggan

	3. Menekan tombol add 4. Memasukkan form pada master data pelanggan 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada master data pelanggan
Hasil	Berhasil menyimpan dan memasukkan data pada master data pelanggan
Status	Valid

6.3.48 Pengujian Validasi Mengubah master data pelanggan

a. Kasus uji mengubah master data pelanggan

Tabel 6.57 Kasus uji mengubah master data pelanggan

Nama Kasus Uji	Kasus Uji mengubah master data pelanggan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar master data pelanggan 3. Menekan tombol edit 4. Memasukkan data yang baru pada form master data pelanggan 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada master data pelanggan
Hasil	Berhasil menyimpan dan mengubah data pada master data pelanggan
Status	Valid

6.3.49 Pengujian Validasi Melihat master data pelanggan

a. Kasus uji melihat master data pelanggan

Tabel 6.58 Kasus uji Melihat master data pelanggan

Nama Kasus Uji	Kasus Uji melihat master data pelanggan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman master data pelanggan
Hasil yang diharapkan	Menampilkan daftar master data pelanggan
Hasil	Menampilkan daftar master data pelanggan

Status	Valid
--------	-------

6.3.50 Pengujian Validasi Menghapus master data pelanggan

- a. Kasus uji menghapus master data pelanggan

Tabel 6.59 Kasus uji menghapus master data pelanggan

Nama Kasus Uji	Kasus Uji menghapus master data pelanggan
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman master data pelanggan 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.51 Pengujian Validasi Membuat master data leasing

- a. Kasus uji Membuat master data leasing

Tabel 6.60 Kasus uji Membuat master data leasing

Nama Kasus Uji	Kasus Uji Membuat master data leasing
Prosedur	1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar master data leasing 3. Menekan tombol add 4. Memasukkan form pada master data leasing 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada master data leasing
Hasil	Berhasil menyimpan dan memasukkan data pada master data leasing
Status	Valid

6.3.52 Pengujian Validasi Mengubah master data leasing

- a. Kasus uji mengubah master data leasing

Tabel 6.61 Kasus uji mengubah master data leasing

Nama Kasus Uji	Kasus Uji mengubah master data leasing
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman daftar master data leasing 3. Menekan tombol edit 4. Memasukkan data yang baru pada form master data leasing 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada master data leasing
Hasil	Berhasil menyimpan dan mengubah data pada master data leasing
Status	Valid

6.3.53 Pengujian Validasi Melihat master data leasing

- a. Kasus uji melihat master data leasing

Tabel 6.62 Kasus uji Melihat master data leasing

Nama Kasus Uji	Kasus Uji melihat master data leasing
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman master data leasing
Hasil yang diharapkan	Menampilkan daftar master data leasing
Hasil	Menampilkan daftar master data leasing
Status	Valid

6.3.54 Pengujian Validasi Menghapus master data leasing

- a. Kasus uji menghapus master data leasing

Tabel 6.63 Kasus uji menghapus master data leasing

Nama Kasus Uji	Kasus Uji menghapus master data leasing
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai sales 2. Masuk ke dalam halaman master data leasing 3. Menekan tombol delete

Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.55 Pengujian Validasi Membuat data kas master

a. Kasus uji Membuat data kas master

Tabel 6.64 Kasus uji Membuat data kas master

Nama Kasus Uji	Kasus Uji Membuat data kas master
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman daftar data kas master 3. Menekan tombol add 4. Memasukkan form pada data kas master 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada data kas master
Hasil	Berhasil menyimpan dan memasukkan data pada data kas master
Status	Valid

6.3.56 Pengujian Validasi Mengubah data kas master

a. Kasus uji mengubah data kas master

Tabel 6.65 Kasus uji mengubah data kas master

Nama Kasus Uji	Kasus Uji mengubah data kas master
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman daftar data kas master 3. Menekan tombol edit 4. Memasukkan data yang baru pada form data kas master 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada data kas master
Hasil	Berhasil menyimpan dan mengubah data pada data kas master

Status	Valid
--------	-------

6.3.57 Pengujian Validasi Melihat data kas master

- a. Kasus uji melihat data kas master

Tabel 6.66 Kasus uji Melihat data kas master

Nama Kasus Uji	Kasus Uji melihat data kas master
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data kas master
Hasil yang diharapkan	Menampilkan daftar data kas master
Hasil	Menampilkan daftar data kas master
Status	Valid

6.3.58 Pengujian Validasi Menghapus data kas master

- a. Kasus uji menghapus data kas master

Tabel 6.67 Kasus uji menghapus data kas master

Nama Kasus Uji	Kasus Uji menghapus data kas master
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data kas master 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.59 Pengujian Validasi Membuat data kas terima

- a. Kasus uji Membuat data kas terima

Tabel 6.68 Kasus uji Membuat data kas terima

Nama Kasus Uji	Kasus Uji Membuat data kas terima
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman daftar data kas terima 3. Menekan tombol add 4. Memasukkan form pada data kas terima

	5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada data kas terima
Hasil	Berhasil menyimpan dan memasukkan data pada data kas terima
Status	Valid

6.3.60 Pengujian Validasi Mengubah data kas terima

a. Kasus uji mengubah data kas terima

Tabel 6.69 Kasus uji mengubah data kas terima

Nama Kasus Uji	Kasus Uji mengubah data kas terima
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman daftar data kas terima 3. Menekan tombol edit 4. Memasukkan data yang baru pada form data kas terima 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada data kas terima
Hasil	Berhasil menyimpan dan mengubah data pada data kas terima
Status	Valid

6.3.61 Pengujian Validasi Melihat data kas terima

a. Kasus uji melihat data kas terima

Tabel 6.70 Kasus uji Melihat data kas terima

Nama Kasus Uji	Kasus Uji melihat data kas terima
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data kas terima
Hasil yang diharapkan	Menampilkan daftar data kas terima
Hasil	Menampilkan daftar data kas terima
Status	Valid

6.3.62 Pengujian Validasi Menghapus data kas terima

- a. Kasus uji menghapus data kas terima

Tabel 6.71 Kasus uji menghapus data kas terima

Nama Kasus Uji	Kasus Uji menghapus data kas terima
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data kas terima 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.63 Pengujian Validasi Membuat data kas keluar

- a. Kasus uji Membuat data kas keluar

Tabel 6.72 Kasus uji Membuat data kas keluar

Nama Kasus Uji	Kasus Uji Membuat data kas keluar
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman daftar data kas keluar 3. Menekan tombol add 4. Memasukkan form pada data kas keluar 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada data kas keluar
Hasil	Berhasil menyimpan dan memasukkan data pada data kas keluar
Status	Valid

6.3.64 Pengujian Validasi Mengubah data kas keluar

- a. Kasus uji mengubah data kas keluar

Tabel 6.73 Kasus uji mengubah data kas keluar

Nama Kasus Uji	Kasus Uji mengubah data kas keluar
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir

	2. Masuk ke dalam halaman daftar data kas keluar 3. Menekan tombol edit 4. Memasukkan data yang baru pada form data kas keluar 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada data kas keluar
Hasil	Berhasil menyimpan dan mengubah data pada data kas keluar
Status	Valid

6.3.65 Pengujian Validasi Melihat data kas keluar

- a. Kasus uji melihat data kas keluar

Tabel 6.74 Kasus uji Melihat data kas terima

Nama Kasus Uji	Kasus Uji melihat data kas keluar
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data kas keluar
Hasil yang diharapkan	Menampilkan daftar data kas keluar
Hasil	Menampilkan daftar data kas keluar
Status	Valid

6.3.66 Pengujian Validasi Menghapus data kas keluar

- a. Kasus uji menghapus data kas keluar

Tabel 6.75 Kasus uji menghapus data kas keluar

Nama Kasus Uji	Kasus Uji menghapus data kas keluar
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data kas keluar 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.67 Pengujian Validasi Melihat riwayat kas terima

- a. Kasus uji melihat riwayat kas terima

Tabel 6.76 Kasus uji Melihat riwayat kas terima

Nama Kasus Uji	Kasus Uji melihat riwayat kas terima
Prosedur	1. Sudah berada dalam kondisi logon sebagai manager 2. Masuk ke dalam halaman data riwayat kas terima
Hasil yang diharapkan	Menampilkan daftar riwayat kas terima
Hasil	Menampilkan daftar riwayat kas terima
Status	Valid

6.3.68 Pengujian Validasi Melihat riwayat kas keluar

- a. Kasus uji melihat riwayat kas keluar

Tabel 6.77 Kasus uji Melihat riwayat kas terima

Nama Kasus Uji	Kasus Uji melihat riwayat kas keluar
Prosedur	1. Sudah berada dalam kondisi logon sebagai manager 2. Masuk ke dalam halaman data riwayat kas keluar
Hasil yang diharapkan	Menampilkan daftar riwayat kas keluar
Hasil	Menampilkan daftar riwayat kas keluar
Status	Valid

6.3.69 Pengujian Validasi Melihat riwayat sales order

- a. Kasus uji melihat riwayat sales order

Tabel 6.78 Kasus uji Melihat riwayat sales order

Nama Kasus Uji	Kasus Uji melihat riwayat sales order
Prosedur	1. Sudah berada dalam kondisi logon sebagai manager 2. Masuk ke dalam halaman data riwayat sales order
Hasil yang diharapkan	Menampilkan daftar riwayat sales order
Hasil	Menampilkan daftar riwayat sales order
Status	Valid

6.3.70 Pengujian Validasi Menambah data antrian servis

a. Kasus uji Menambah data antrian servis

Tabel 6.79 Kasus uji Menambah data antrian servis

Nama Kasus Uji	Kasus Uji Menambah data antrian servis
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai mekanik 2. Masuk ke dalam halaman daftar antrian servis 3. Menekan tombol add 4. Memasukkan form pada data antrian servis 5. Menekan tombol simpan
Hasil yang diharapkan	Berhasil menyimpan dan memasukkan data pada antrian servis
Hasil	Berhasil menyimpan dan memasukkan data pada antrian servis
Status	Valid

6.3.71 Pengujian Validasi Mengubah data antrian servis

a. Kasus uji mengubah data antrian servis

Tabel 6.80 Kasus uji mengubah data antrian servis

Nama Kasus Uji	Kasus Uji mengubah data antrian servis
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai mekanik 2. Masuk ke dalam halaman daftar data antrian servis 3. Menekan tombol edit 4. Memasukkan data yang baru pada form data antrian servis 5. Menekan tombol update
Hasil yang diharapkan	Berhasil menyimpan dan mengubah data pada data antrian servis
Hasil	Berhasil menyimpan dan mengubah data pada data antrian servis
Status	Valid

6.3.72 Pengujian Validasi Melihat data antrian servis

- a. Kasus uji melihat data antrian servis

Tabel 6.81 Kasus uji Melihat data antrian servis

Nama Kasus Uji	Kasus Uji melihat data antrian servis
Prosedur	1. Sudah berada dalam kondisi logon sebagai kasir 2. Masuk ke dalam halaman data antrian servis
Hasil yang diharapkan	Menampilkan daftar data antrian servis
Hasil	Menampilkan daftar data antrian servis
Status	Valid

6.3.73 Pengujian Validasi Menghapus data antrian servis

- a. Kasus uji menghapus data antrian servis

Tabel 6.82 Kasus uji menghapus data antrian servis

Nama Kasus Uji	Kasus Uji menghapus data antrian servis
Prosedur	1. Sudah berada dalam kondisi logon sebagai mekanik 2. Masuk ke dalam halaman data antrian servis 3. Menekan tombol delete
Hasil yang diharapkan	Data berhasil terhapus
Hasil	Data berhasil terhapus
Status	Valid

6.3.74 Pengujian Validasi Merubah status antrian servis

- a. Kasus uji Merubah status antrian servis

Tabel 6.83 Kasus uji Merubah status antrian servis

Nama Kasus Uji	Kasus Uji Merubah status antrian servis
Prosedur	1. Sudah berada dalam kondisi logon sebagai mekanik 2. Masuk ke dalam halaman data antrian servis 3. Menekan tombol selesai
Hasil yang diharapkan	Data berhasil berubah menjadi selesai antri
Hasil	Data berhasil berubah menjadi selesai antri
Status	Valid

6.3.75 Pengujian Validasi Melakukan pengecekan gratis servis

a. Kasus uji Melakukan pengecekan gratis servis

Tabel 6.84 Kasus uji Melakukan pengecekan gratis servis

Nama Kasus Uji	Kasus Uji Melakukan pengecekan gratis servis
Prosedur	<ol style="list-style-type: none"> 1. Sudah berada dalam kondisi logon sebagai mekanik 2. Masuk ke dalam halaman pengecekan gratis servis 3. Memasukan data nomer rangka dan nomer mesin
Hasil yang diharapkan	Sistem akan mengeluarkan apakah motor tersebut masih memiliki gratis servis
Hasil	Sistem akan mengeluarkan apakah motor tersebut masih memiliki gratis servis
Status	Valid

6.4 Pembahasan Hasil Pengujian

Dari hasil pengujian pada sistem yang telah dibangun, dilakukan analisis terhadap hasil pengujian. Analisis pengujian ini dilakukan untuk mengetahui apakah sistem yang dibangun dapat berjalan dengan baik tanpa adanya gangguan.

6.4.1 Analisis Pengujian Unit

Pengujian unit pada sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo dilakukan dengan metode basis path dengan mengambil 3 sampel operasi antara lain operasi LockInvoice(), PrintDeliveryOrder(), SearchNoInvoiceWithTagihan(). Masing-masing dari operasi tersebut memiliki nilai *cyclomatic complexity* yaitu 2, 4 dan 2 yang menunjukkan bahwa kompleksitas dari tiga unit tersebut rendah. Sehingga dapat disimpulkan bahwa program sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo mudah untuk di maintenance

6.4.2 Analisis Pengujian Integrasi

Pengujian integrasi pada sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo dilakukan dengan metode basis path dengan mengambil 3 sampel fungsi. Operasi pertama adalah operasi sendPermintaanBarangKePusat() pada klas EmailController yang memanggil operasi send() dan failure() pada klas Mail. Operasi kedua adalah operasi searchNoPo() pada klas AutoComplateController yang memanggil operasi get() pada klas Request. Operasi ketiga adalah operasi cronTaskUltah() pada klas CrmDashboardController yang memanggil operasi api_wa_send_plain() pada klas

CrmDashboardController. Masing-masing dari fungsi tersebut memiliki nilai cyclomatic complexity yaitu 2, 2 dan 2 yang menunjukkan bahwa kompleksitas dari tiga integrasi tersebut rendah. Sehingga dapat disimpulkan bahwa program sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo mudah untuk di maintenance.

6.4.3 Analisis Pengujian Validasi

Hasil pengujian validasi menyatakan bahwa delapan puluh satu fungsionalitas yang terdapat pada sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo telah dilakukan pengujian dan memperoleh hasil valid atau dapat berjalan dengan baik sesuai fungsinya, baik kondisi normal maupun alternatif.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian sistem yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan, sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo ini memiliki 81 kebutuhan fungsional. Untuk penentuan kebutuhan fungsional didapatkan dari hasil survei terhadap karyawan yang bekerja di PT. Mega Abadi Motorindo dengan permasalahan pada bidang penjualan dan pelayanan pelanggan. Dalam penelitian ini, digunakan pemodelan untuk memudahkan pengembang untuk memahami sistem, pemodelan yang digunakan adalah use case diagram dan use case scenario.

2. Tahap Perancangan yang dilakukan menurut hasil analisis kebutuhan dibagi menjadi beberapa proses. Diantaranya adalah perancangan data, dan perancangan komponen serta perancangan antarmuka. Lalu dilakukan juga pemodelan *sequence diagram* dan kelas diagram pada tahap perancangan. Setelah itu implementasi sistem dilakukan dengan mengimplementasikan hasil perancangan yang sudah dilakukan sebelumnya. Implementasi program dilakukan dengan menggunakan bahasa pemrograman PHP dan menggunakan konsep MVC pada framework Laravel. Implementasi basis data menggunakan basis data MySQL dengan server phpMyAdmin. Sedangkan pada implementasi antarmuka dibuat dengan menggunakan *bootstrap* serta penambahan jQuery dan ajax agar mempermudah penggunaan sistem.

3. Tahap pengujian yang dilakukan setelah melakukan perancangan dan implementasi dari sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo meliputi metode whitebox serta blackbox. Pengujian whitebox yang dilakukan yaitu pengujian unit dan integrasi. Pengujian whitebox tersebut diambil sampel 6 fungsi pada sistem. Sedangkan pengujian blackbox dilakukan pengujian validasi untuk memastikan bahwa kebutuhan yang ada pada sistem telah sesuai dengan apa yang diharapkan. Dari pengujian fungsional diperoleh bahwa kompleksitas program rendah sehingga dapat dengan mudah dilakukan maintenance pada program dan 100% valid atau tidak ada kesalahan pada sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem Aplikasi Penjualan dan Pelayanan Pelanggan pada PT. Mega Abadi Motorindo ini selanjutnya antara lain:

1. Penambahan fungsi untuk pelaporan langsung kepada manager secara *realtime* menggunakan email ataupun *push notification*.
2. Dilakukan pengembangan dengan menggunakan aplikasi android, agar sistem dapat diakses dengan mudah dan cepat dimana saja.



DAFTAR REFERENSI

- Juwitasari, H, Martani, M., & Putra, A. N. G., 2015. Analisis Sistem Informasi Akuntansi Pembelian dan Persediaan pada PT. XYZ [online] tersedia di < <http://journal.binus.ac.id/index.php/comtech/article/view/2294/1713>>
- Komputer W. 2010. Panduan Belajar MySQL Database Server [online] tersedia di < <https://books.google.co.id/books?id=rR1VNYpc08wC>>
- Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. Jurnal Teknologi Informasi Dan Ilmu Komputer, 5(1), 77. <https://doi.org/10.25126/jtiik.201851610>
- Luhukay,D., Kurniawan, Y., & Titan, 2013. Analisis Perancangan Sistem Informasi Penjualan dan Persediaan pada PT. XYZ [online] tersedia di < <http://journal.binus.ac.id/index.php/comtech/article/view/2696/2102>>
- Muhammd Suleman Sarwar, Mir & Shahzad, Sara & Ahmad, Ibrar. (2013). Cyclomatic complexity: The nesting problem. 8th International Conference on Digital Information Management, ICDIM 2013. 274-279. 10.1109/ICDIM.2013.6693981.
- Nugroho A. 2009. Rekayasa Perangkat Lunak Menggunakan UML dan Java [online] tersedia di < https://books.google.co.id/books?id=uly0ekGR_X4C>
- Putratama V., Supono. 2016. Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter [online] tersedia di < <https://books.google.co.id/books?id=QtPODAAAQBAJ>>
- Singh S. K. 2009. Database Systems [online] tersedia di < <https://books.google.co.id/books?id=8PNCKe2SpRwC>>
- Sommerville, Ian. 2011. Software Engineering (Rekayasa Perangkat Lunak). Jakarta: Erlangga
- Stauffer M. 2016. Laravel: Up and Running [online] tersedia di < <https://books.google.co.id/books?id=-IGRDQAAQBAJ>>
- Sugiarti, Yuni, 2013. Analisis dan Perancangan UML (Unified Modeling Language), Graha Ilmu. Yogyakarta
- Pressman R. S. 2014. Software Engineering: A Practitioner's Approach